TouchPAD

TPD/VPD Series HMI Device

User Manual Version 1.0.26

ICP DAS Co., Ltd.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright @ 2015 by ICP DAS Co., Ltd. All rights are reserved.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Support

ICP DAS takes your problem as ours.

If you have any problem, please feel free to contact us.
You can count on us for quick response.

Email: service@icpdas.com

Tel: 886-3-5973336

Also, the FTP site of ICP DAS has contents about TouchPAD which you may be interested in. We believe that those contents may be helpful to your work.

FTP: ftp://ftp.icpdas.com/pub/cd/touchpad/

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Preface

Thank you for buying TPD/VPD Series HMI Devices, TouchPADs, which are made by ICP DAS Co., Ltd. We suggest you read through this user manual before you set up these devices and develop their programs.

Purpose

- This manual shows how to use TouchPADs and develop programs.
- This manual mainly contains the following parts:
  - Introduction: basic understandings of TouchPADs.
  - Hardware: specifications, dimensions, and installations.
  - Software: mainly how to build a project and HMIWorks introductions.

Personnel

This manual is fit for following personnel:

- End Users
- Engineers
- Technicians

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Table of Contents

Preface 3
1. Introduction 7
1.1. Advanced Features 7
1.2. Applications of TouchPAD 8
1.3. All Kinds of Situations 12
2. General Specifications 13
2.1. TPD Series 13
2.2. VPD Series 14
3. Hardware 16
3.1. Hardware Appearance 16
3.2. Installation and Wiring 17
4. Set up Devices and Connect to I/O 20
4.1. Preparation 20
4.1.1. Language Options 20
4.2. Setup Devices 20
4.2.1. Setup RS-485 Downloaded devices (TPD-280, TPD-280-Mx, TPD-280-H) 21
4.2.2. Setup Ethernet-downloaded devices (TPD-283, TPD-283-Mx, TPD-283-H, TPD-283U-Mx, TPD-283U-H, TPD-433-Mx, TPD-433-H, TPD-703, TPD-703-64) 23
4.2.3. Setup USB-downloaded devices (TPD-28xU, TPD-43x, TPD-43xF, VPD-13x, VPD-14x, VPD-133-H) 30
4.2.4. Calibrations 33
4.3. Connecting to I/O Devices 33
4.4 Connecting to I/O Expansion Boards (XVBoard) 35
5. Development Software, HMIWorks 38
5.1. The Construction of HMIWorks 39
5.2. The Options of TouchPAD 41
5.2.1 Project configurations 41
5.2.2 Language Options 43
5.3. Ladder Designer 43
5.3.1. Getting Started 44
5.3.2. Introduction to Ladder Designer 44
5.3.3. Operations of Ladder Designer 50

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.3.4. User-Defined Function Block 64
5.3.5. Associate Tags with Tools 67
5.3.6. User-Defined I/O Modules 70
5.3.1. Data exchange 77
5.4. Frames and Components 80
5.4.1. Commons of Components and Frames 81
5.4.2. Frame 91
5.4.3. Rectangle 92
5.4.4. Ellipse 92
5.4.5. Text 93
5.4.6. Picture 94
5.4.7. Line 96
5.4.8. TextPushButton 96
5.4.9. Slider 99
5.4.10. BitButton 100
5.4.11. HotSpot 100
5.4.12. CheckBox 101
5.4.13. Label 101
5.4.14. RadioButton 104
5.4.15. Timer 106
5.4.16. PaintBox 107
5.4.17. ObjectList 108
5.5. Menus 113
5.5.1. Cascading and Grouping, Arrange Menu 113
5.5.2. Rotating and Flipping, Edit Menu 116
5.5.3. Frame Managing and Aligning, Layout Menu 118
5.5.4. Build and Download to Run, Run Menu 121
5.5.5. Library Management, Popup Menu 122
6. Making a Simple Project 127
6.1. Your First Project Using Standard C 127
6.2. Your First Project Using Ladder 130
6.3. Integrating TPD-280 Series with I/O modules 135
6.4. Integrating TPD-283 Series with I/O modules 138
7. Advanced Programming in C 142
7.1. Adding a New File to Project 142
7.2. Updating Properties in Run Time 143
7.2.1. FillColor and Text of a TextPushButton 143
7.2.2. Percentage of a Slider 145

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

7.2.3. Selected of a CheckBox 146
7.2.4. Font, Text and TextColor of a Label 148
7.3. Accessing Tags in Ladder 150
Appendix 152
A. FAQ 152
A.1. What to do if screen flashes? 152
A.2. How to have higher-resolution Picture? 152
A.3. How does a TouchPAD control I/O? 152
A.4. How to change Font of Text? 152
A.5. How to represent decimals for Ladder Designer? 153
A.6. How to remove the startup beep of a TPD-283 device? 153
A.7. How to customize the generated code? 153
A.8. How to store data in the flash? 154
A.9. How to use soft reset? 154

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

1. Introduction

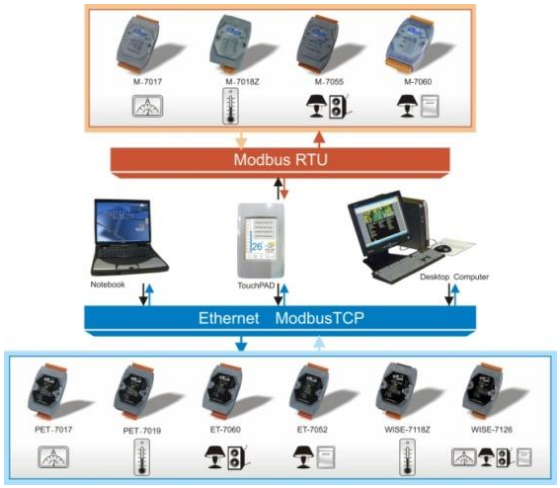
Our solution for HMI (Human Machine Interface) is composed of GUI (Graphical User Interface) based touch screens and an integrated software development package. ICP DAS hears the voices of our customers and is dedicated to providing a series of solutions particularly for intelligent building, equipment monitoring, factory automation and automatic controls. Its development software, HMIWorks, provides plenty of widgets and a variety of templates. Combined with the high resolution color touch screen of the TouchPAD series, a GUI can be realized with your own unique fashion and style. Development is no longer difficult and project accomplishment is within reach.

ICP DAS provides two types of touch HMI devices, the TPD series and the VPD series. The TPD series is designed for home/building automation applications and the VPD series is designed for factory/machine automation applications. Both have many common features, such as a high-resolution touch screen, RTC, and a variety of communication interfaces, including RS-232/RS-485, Ethernet, USB. However, each still has its own specific features for its respective target applications. For the TPD series, you can use an external wall box to help you smoothly blend the TPD series device into your decoration. For the VPD series, the rubber keypad, IP-65 waterproof front panel and DIN-Rail/panel mounting are designed for harsh environment, and are especially suitable for factories.

1.1. Advanced Features

- Excellent C/P ratio (cost/performance)
- Workable under tough environments, operating temperature: -20℃~70℃
- High Color resolution touch screen
- PoE, Power over Ethernet (TPD-283)
- RS-485 network (TPD-280/TPD-280U/TPD-430/VPD-130)

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



- GUI design
- Complete and powerful development tool, easy integration with touch HMI devices, quick design for a variety of applications
- Supports the popular C programming language
- Ladder logic design
- IP rating : TPD series – IP40, VPD- series – IP65
- Support data exchange function.
- Free Developer software, HMIWorks

1.2. Applications of TouchPAD

Interactions between human and machine are getting more and more important since automation control has emerged. From systematic surveillance of equipments, status monitoring of house appliances or even measurements of temperature and humidity, HMI devices play an indispensable role in passing information. In the early time, HMI devices consist of lights, meters, 7-segment display. For now, LED and LCD are prevalently used and ICP DAS releases TouchPAD as a state-of-the-art solution.

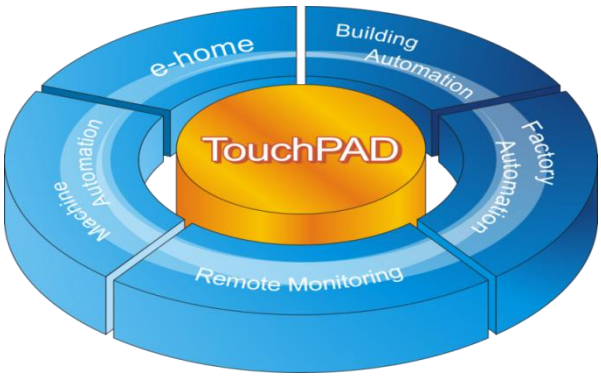
In addition to GUI and touch LCD, the solution of ICP DAS provides development software tool, HMIWorks. Besides, it has plenty of widgets and varieties of templates, and so it's easy to present customers with professional interface with modern styles. Moreover, HMIWorks supports C language. It makes easy timing control and logic design and in turns makes TouchPAD more powerful. Through standard communication protocols and SCADA

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

(Supervisory Control and Data Acquisition) software, it is directly inoculated with background software and attains the best effect of completely integration.

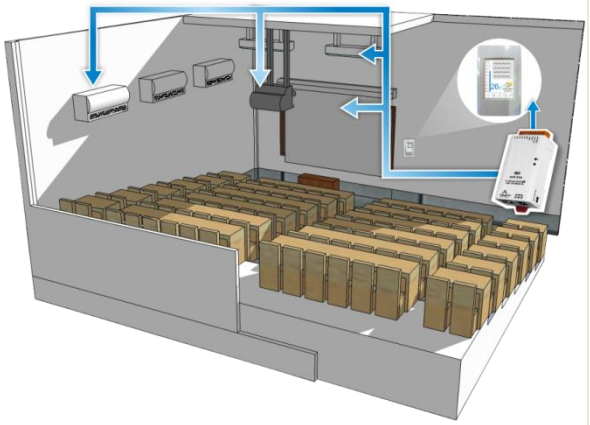
Below are figures of the application fields of TouchPAD, intelligent building and classroom automation. In these examples, TouchPADs are used to control lights, curtains, air conditioners, stereos, projectors, projector screens, and to monitor temperature, humidity, and weather conditions.

Application Fields of TouchPAD

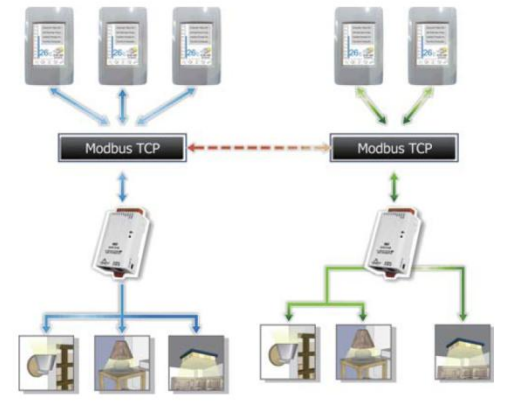


Intelligent Building Example

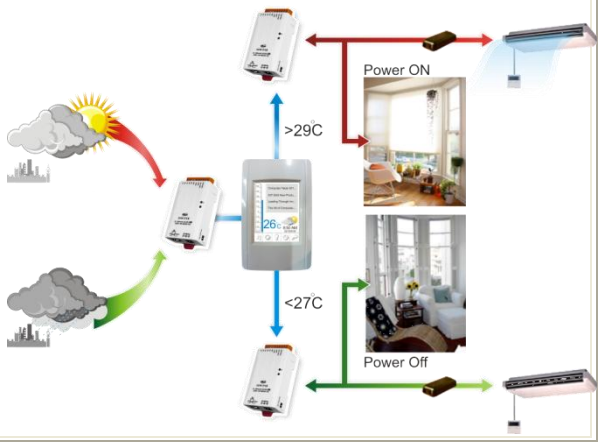
Classroom Automation Example



Single-way, Two-way, Multi-way Switches

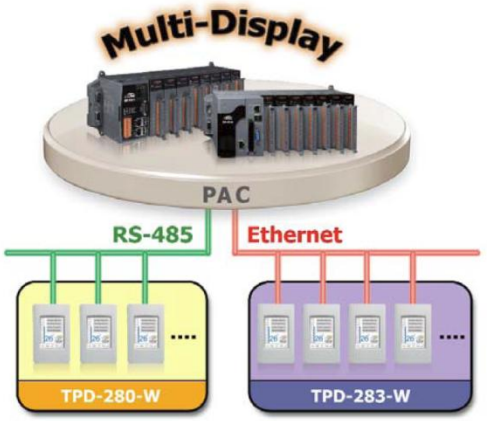


Temperature Control Example

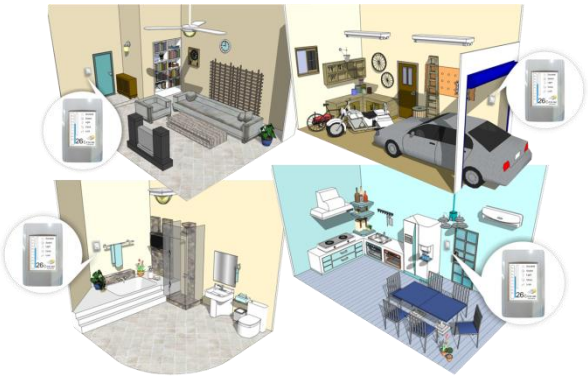


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Multi-Display

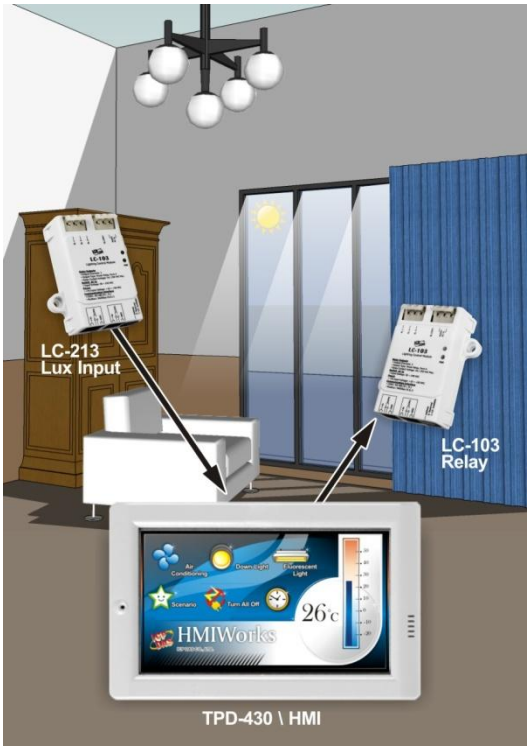


TouchPAD Anywhere

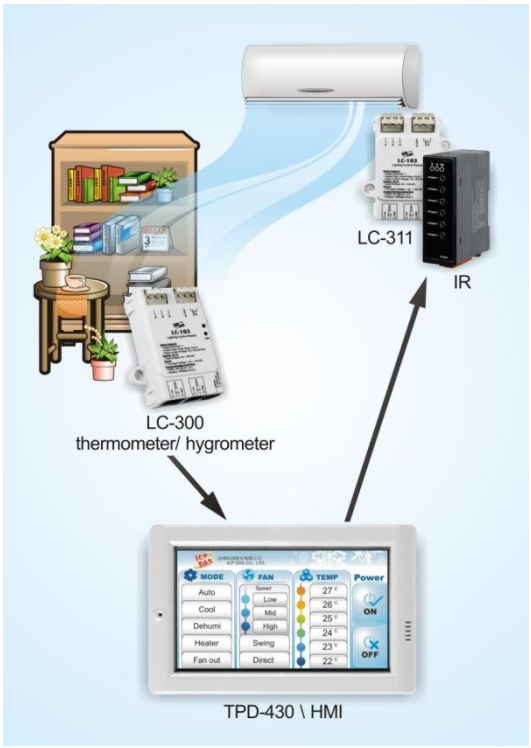


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Electrical Curtain

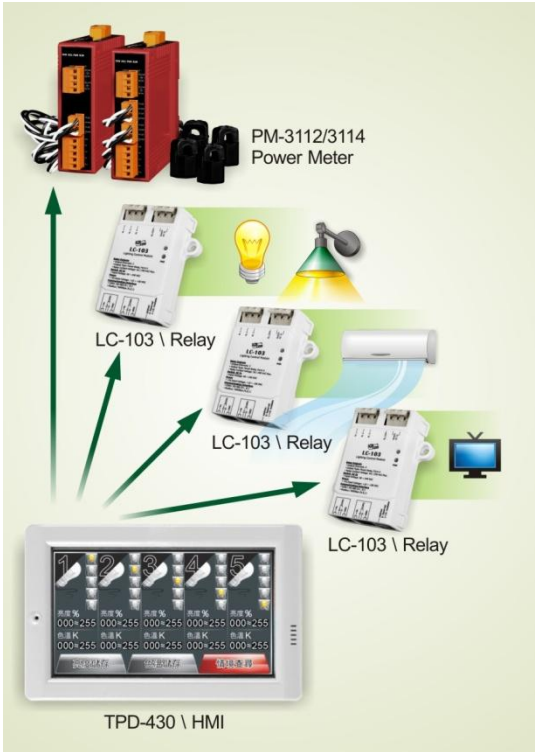


Integration of Infrared appliances



Display/Control for Little Machinery

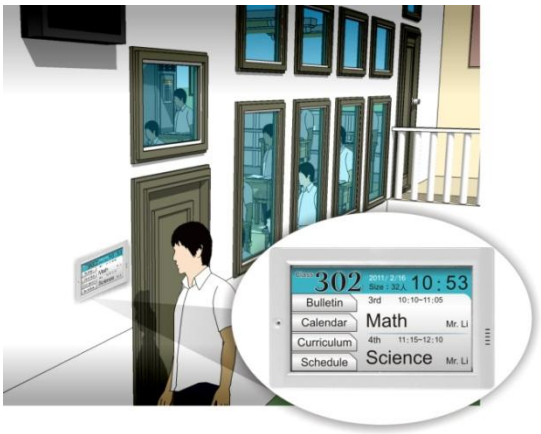
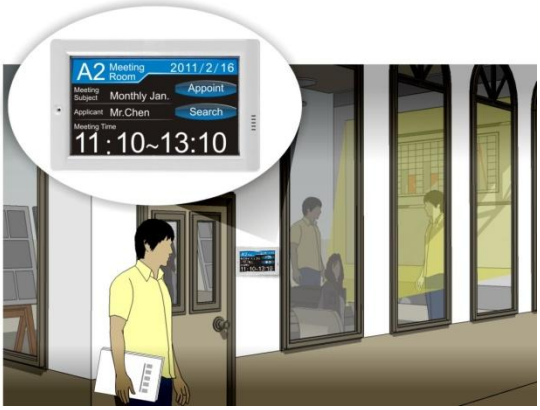
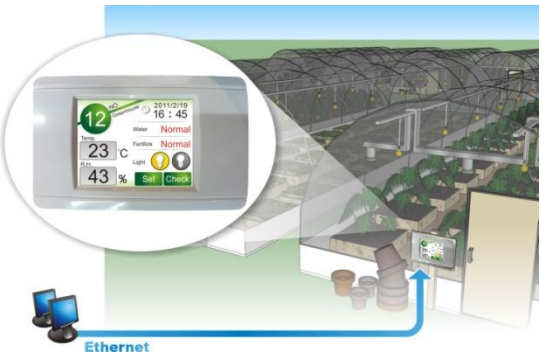
Power Management



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

1.3. All Kinds of Situations

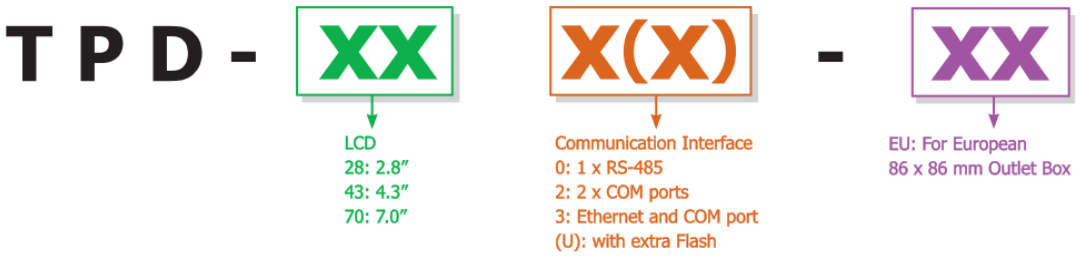
1.4. TouchPAD can be applied in all kinds of situations you even have imagined. In these situations, TouchPAD is used as a window for signal transmissions and arranges incoming messages.

In the hotel, on the wall 	In the school, in the classroom 
In the office, outside the meeting room 	In the farm, in the front door of the greenhouse 

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

2. General Specifications

2.1. TPD Series



2.8" (Resolution: 240 x 320)

Model Name	Extra Flash	Image Storage Capacity	Ethernet	COM port	RTC	Outlet Box	External Wall Box	Power Input
TPD-280	-	1	-	1x RS-485	-	OB120	EWB-T28	+10 ~ 30 VDC
TPD-283		1	Yes	-				PoE
TPD-280U	8 MB	54	-	1x RS-485	Yes			+10 ~ 30 VDC
TPD-283U	8 MB	54	Yes	1x RS-485				+10 ~ 30 VDC or PoE

2.8" (Resolution: 240 x 320)

Model Name	Extra Flash	Image Storage Capacity	Ethernet	COM port	RTC	Outlet Box	External Wall Box	Power Input
TPD-280-M1/M2/M3	-	4	-	1x RS-485	-	OB120	EWB-T28	+12 ~ 48 VDC
TPD-283-M1/M2/M3		4	Yes	-	-			PoE
TPD-283U-M1/M2/M3	16 MB	108	Yes	1x RS-485	Yes			+12 ~ 48 VDC or PoE

2.8" (Resolution: 240 x 320)

Model Name	Extra Flash	Image Storage Capacity	Ethernet	COM port	RTC	Outlet Box	External Wall Box	Power Input
TPD-280-M1/M2/M3	-	4	-	1x RS-485	-	OB120	EWB-T28	+12 ~ 48 VDC
TPD-283-M1/M2/M3		4	Yes	-	-			PoE
TPD-283U-M1/M2/M3	16 MB	108	Yes	1x RS-485	Yes			+12 ~ 48 VDC or PoE

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4.3" (Resolution: 480 x 272)

Model Name	Extra Flash	Image Storage Capacity	Ethernet	COM port	RTC	Outlet Box	External Wall Box	Power Input
TPD-430	8 MB	32	-	1x RS-485	Yes	OB120	EWB-T43	+10 ~ 30 VDC
TPD-433			Yes	1x RS-485				+10 ~ 30 VDC or PoE
TPD-432F			-	2x RS-485		OB140F OB140FP	EWB-T43F	+10 ~ 30 VDC
TPD-433F			Yes	1x RS-232 1x RS-485				+10 ~ 30 VDC or PoE
TPD-430-EU			-	1x RS-485		-	-	+10 ~ 30 VDC
TPD-433-EU			Yes	1x RS-485				+10 ~ 30 VDC or PoE

7" (Resolution: 800 x 480)

Model Name	Extra Flash	Image Storage Capacity	Ethernet	COM port	RTC	Outlet Box	External Wall Box	Power Input
TPD-703	16 MB	18	Yes	1x RS-232 1x RS-485	Yes	OB170	EWB-T70	+12~48 VDC or PoE
TPD-703-64	64 MB	84						

Note1: Image Storage Capacity depends on the content of program and the size of images. The number is counted by how many images in full screen size can be stored on the device.

Note2: PoE (Power over Ethernet) specification: IEEE 802.3af, Class 1, 48 V

2.2. VPD Series

VPD-X(X)

- X: Form Factor
  - 1: 103 × 103 mm Panel Mount
- X: Touch Screen Size
  - 3: 3.5 inch
  - 4: 4.3 inch
- X(X): Communication Interface
  - 0: RS-485
  - 2: RS-232/RS-485 + RS-485
  - 3: RS-232/RS-485 + RS-485 + Ethernet
  - (N): No Rubber Keypad

VPD Model	LCD	Extra Flash	Image Storage Capacity	Com Port	Ethernet	RTC	Expansion I/O Boards	Rubber Keypad	Ingress Protection	Power Input
VPD-130	3.5" TFT (Resolution 320 x 240)	8 MB	54	RS-232/RS-485	-	Yes	-	Yes	Front Panel: IP65	+12 ~ 48 VDC
VPD-130N							-	-		
VPD-132				COM1: RS-485 or RS-232 COM2: RS-485			Yes	Yes		
VPD-132N								-		
VPD-133					Yes			Yes		+12 ~ 48 VDC or PoE
VPD-133N								-		
VPD-142	4.3" TFT (Resolution 480 x 272)	8 MB	32	COM1: RS-485 or RS-232 COM2: RS-485 or RS-232	-	Yes	Yes	Yes	Front Panel: IP65	+12 ~ 48 VDC
VPD-142N								-		
VPD-143					Yes			Yes		+12 ~ 48 VDC or PoE
VPD-143N								-		

Note1: Image Storage Capacity depends on the content of program and the size of images. The number is counted by how many images in full screen size can be stored on the device.

Note2: PoE (Power over Ethernet) specification: IEEE 802.3af, Class 1, 48 V

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Note:

Communication interface that is only for run time supports the following protocols:

I. For the case of RS-485, Modbus RTU Master and DCON Protocol Master (for ICP DAS I-7000 series modules) are supported.
   We provide API functions to open com port for sending/receiving strings through RS-485.

II. For the case of Ethernet, Modbus TCP Master is supported.
   We provide API functions to sending/receiving strings through TCP.

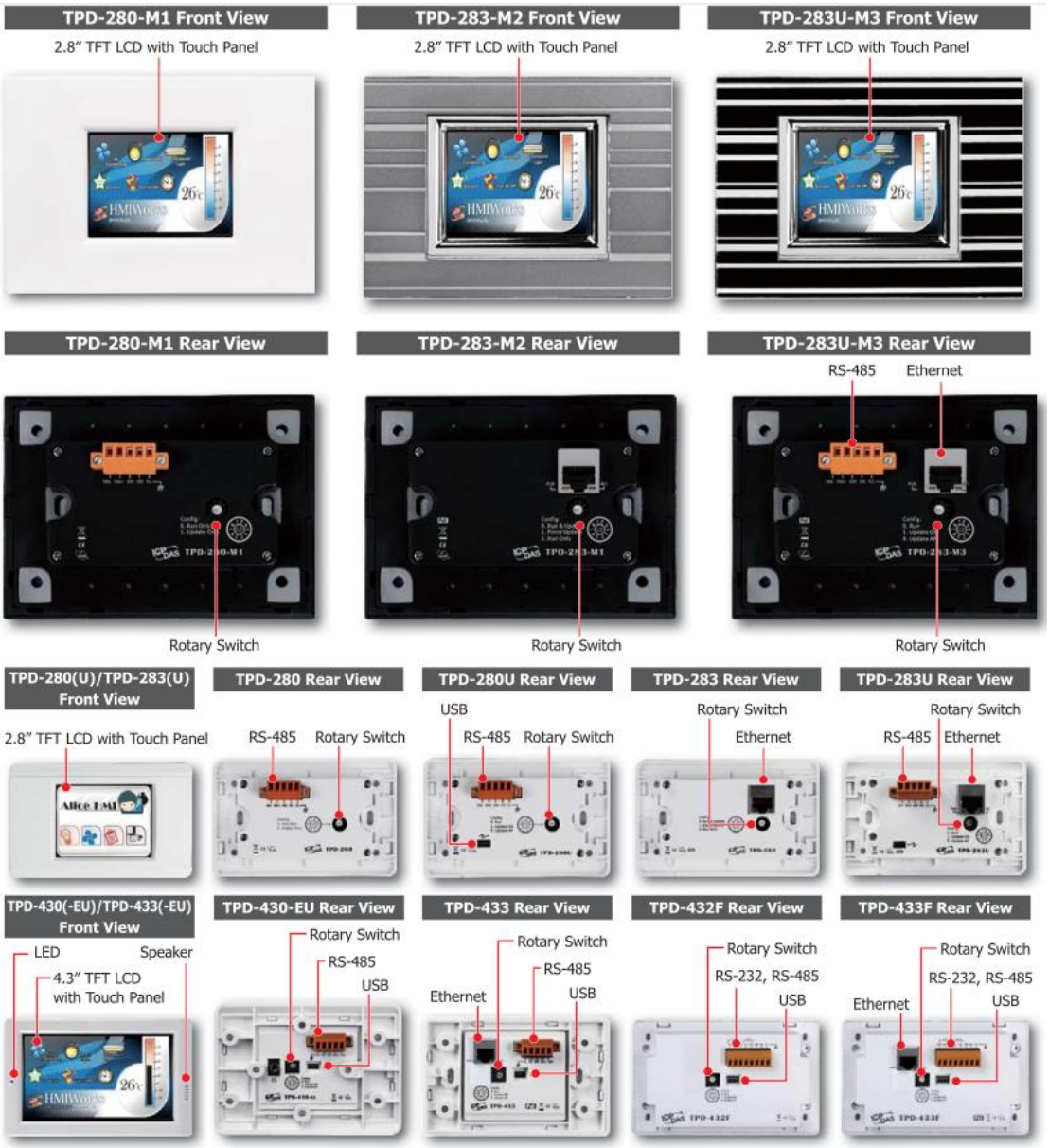
USB is used for firmware update only.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

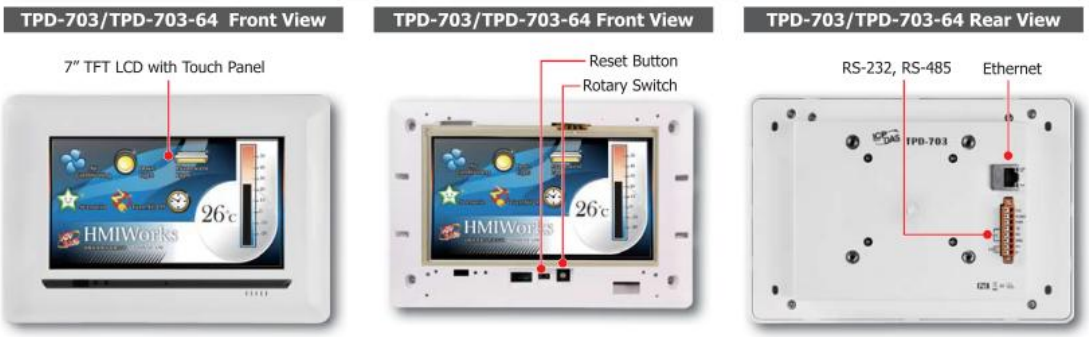
3. Hardware

This chapter shows the overviews, dimensions, etc. of the TouchPAD devices.

3.1. Hardware Appearance



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Important Notice

The reset button does not behave as a reboot to the whole system, only resets the microcontroller and this is not enough to make the program downloading successful. Always cut the power then turn it on before downloading programs.

3.2. Installation and Wiring

Mount the hardware

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

For 4.3" (TPD-432F, TPD-433F)

EWB-T43F (External Wall Box)

OB140F (Outlet Box)

OB-140FP (Outlet Box)

For 4.3" (TPD-430-EU, TPD-433-EU)

Mounted to European-standard 86 mm x 86 mm outlet box.

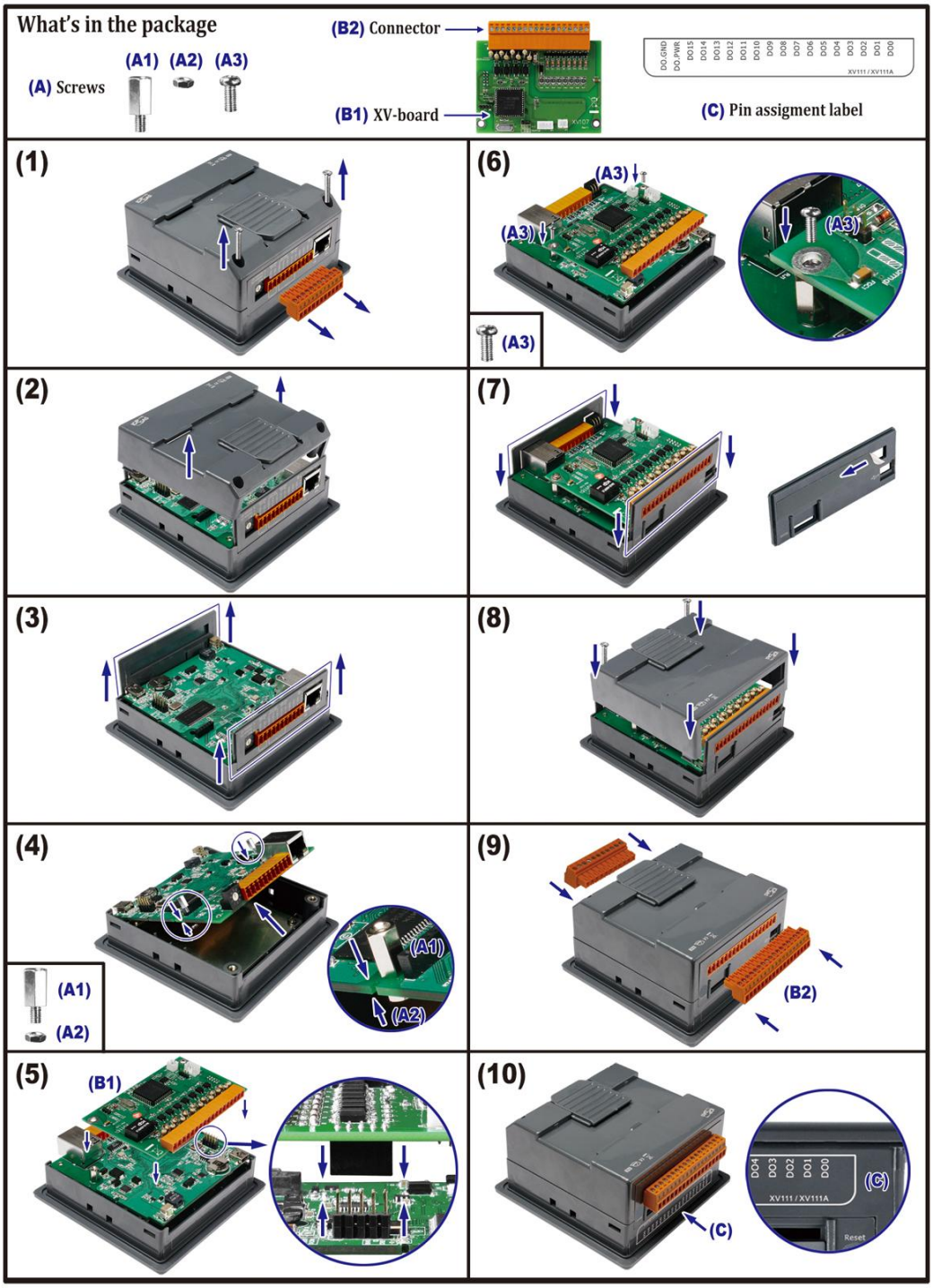
For 7" (TPD-703, TPD-703-64)

EWB-T70 (External Wall Box)

OB170 (Outlet Box)

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

XV-board assembly on VPD-133/VPD-132 2013/02/08



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4. Set up Devices and Connect to I/O

This chapter is divided into two parts. One is setup TouchPAD and the other is connecting TouchPAD to I/O modules.

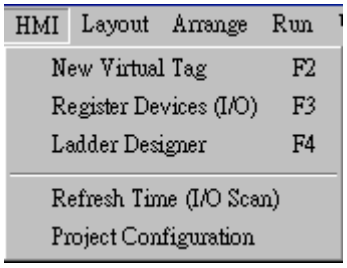
4.1. Preparation

First of all, you should install the HMIWorks development software on your PC. HMIWorks is the development tools for the TouchPAD devices.

Follow the steps as suggested by the HMIWorks setup wizard to finish the installation.

4.1.1. Language Options

You can change EN or ZH-TW
At View > language optons

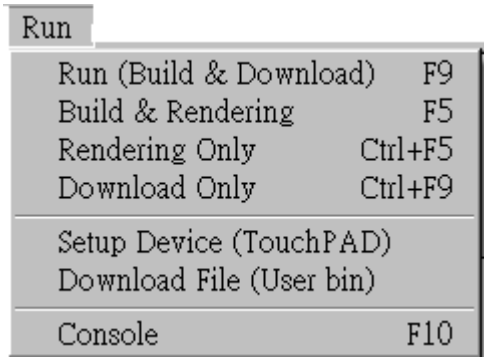


4.2. Setup Devices

The TouchPAD devices are divided into several groups. Before downloading programs to the TouchPAD device, the TouchPAD device must be set up. We describe how to set up each kind of the TouchPAD devices in the following sections.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Click the “**Setup Device (TouchPAD)**” option from the “**Run**” menu of the HMIWorks software to setup TouchPAD.



4.2.1. Setup RS-485 Downloaded devices

(TPD-280, TPD-280-Mx, TPD-280-H)

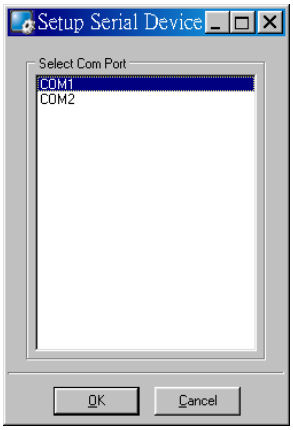
On the back panel of the TouchPAD RS-485 download device, configuration modes can be found. Use the rotary switch to set the configuration mode.

No	Mode	Description
0	Run Only	This mode is used for running programs.
1	Update Only	This mode is used for updating programs.

The TouchPAD RS-485 download device use the RS-485 network as its communication method. It is not able to connect through the Ethernet network.

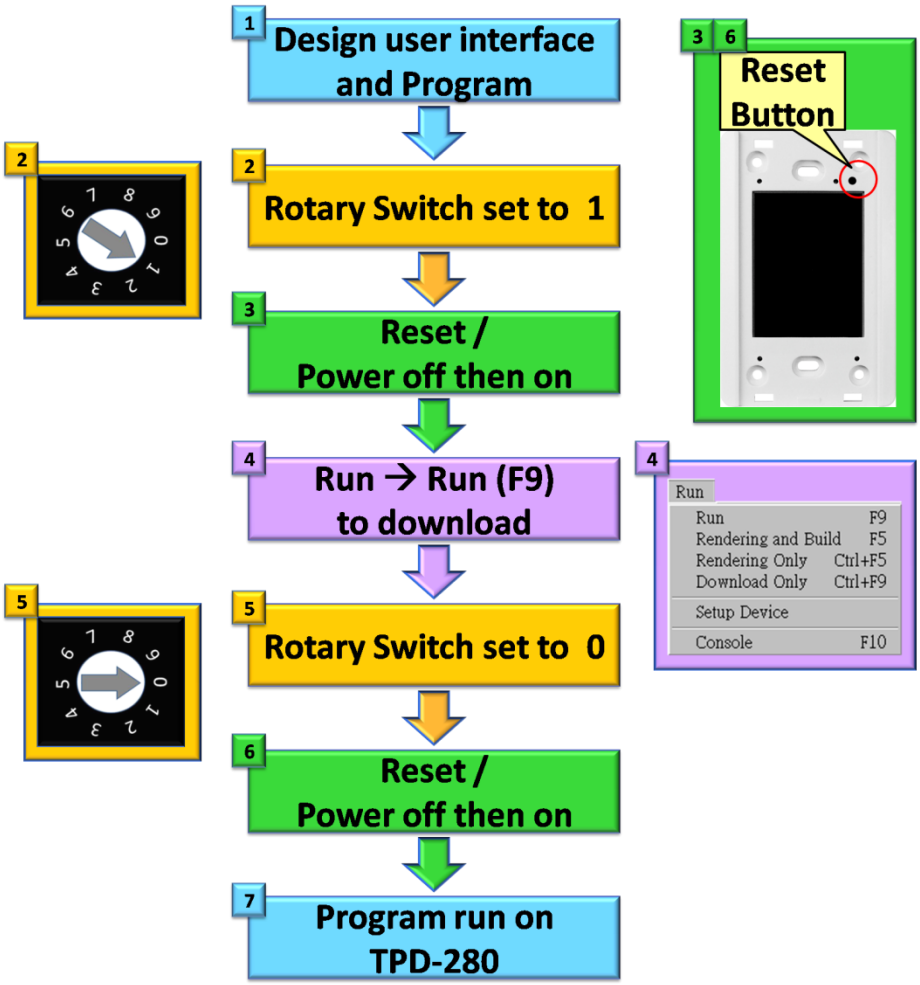
Select the com port that connects to the TouchPAD device.
Before downloading programs to the TouchPAD device, be sure to set up the TouchPAD device to connect to it first.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Flow diagram

Click on the “**Run**” option or the “**Download Only**” option from the “**Run**” menu. As shown in the figure below, set the rotary switch to “1” when downloading the program and set the rotary switch back to “0” when finishing downloading and let the program run.



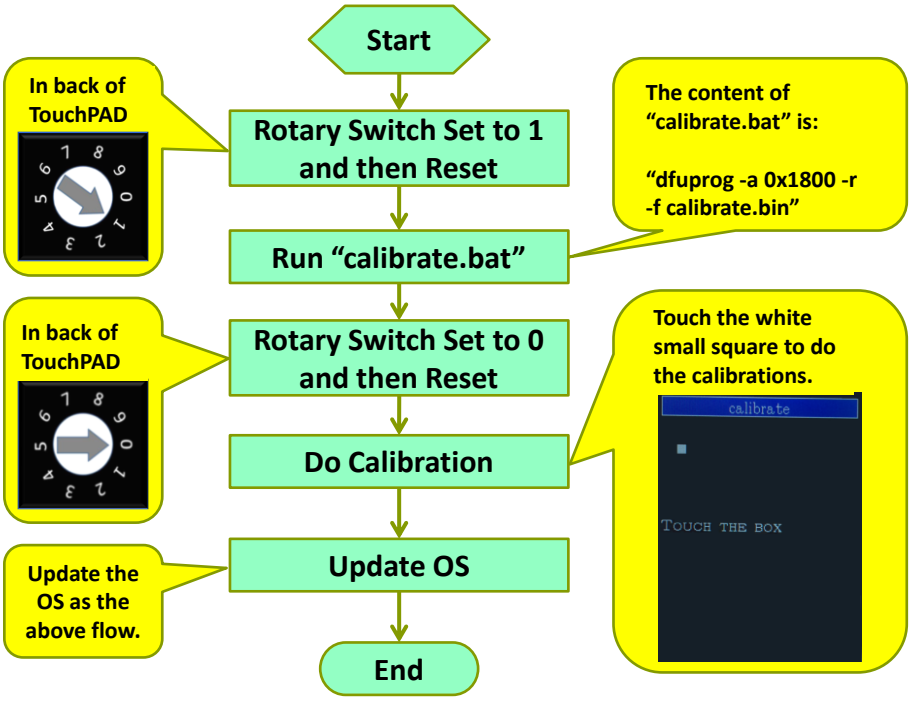
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4.2.2. Setup Ethernet-downloaded devices (TPD-283, TPD-283-Mx, TPD-283-H, TPD-283U-Mx, TPD-283U-H, TPD-433-Mx, TPD-433-H, TPD-703, TPD-703-64)

On the back panel of a TouchPAD Ethernet-download device, configuration modes can be found. Use the rotary switch to set the configuration mode.

No	Mode	Description
0	Run & Update	This is a special run mode which is used in the development stage. The TouchPAD devices can be updated by a PC from the remote side through Ethernet.
1	Force Update	While the application run on the TouchPAD device seriously crashes, use this mode to update an new application to the TouchPAD device.
2	Run Only	Simply run, a TouchPAD device cannot be updated in this mode.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com


Start
In back of TouchPAD
Rotary Switch Set to 1 and then Reset
The content of "calibrate.bat" is:
"dfuprog -a 0x1800 -r -f calibrate.bin"
Run "calibrate.bat"
In back of TouchPAD
Rotary Switch Set to 0 and then Reset
Touch the white small square to do the calibrations.
calibrate
TOUCH THE BOX
Do Calibration
Update the OS as the above flow.
Update OS
End

Default IP settings of the TouchPAD Ethernet-download device.

Item	Value
IP	192.168.255.1
Type	Static IP

The IP settings of a TPD Ethernet device can be classified in one of three types:

Type	Description
Static IP	The TPD Ethernet device has a static, unchangeable IP address.
DHCP	The TPD Ethernet device is assigned an IP address from the DHCP server every time it is turned on (reset).
Runtime Setting	The IP address for the TPD Ethernet device is left unassigned during the building process and is specified in the runtime. We provide API functions for specifying IP settings as well as demos to do that.

Users can change the type of IP settings by downloading a new program into the TPD Ethernet device.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

1. Click the “**Setup Device**” option from the “**Run**” menu, then click the **“Search TouchPad …”** button in the displayed “**Setup Ethernet Device**” window to find the TouchPAD device on the network.

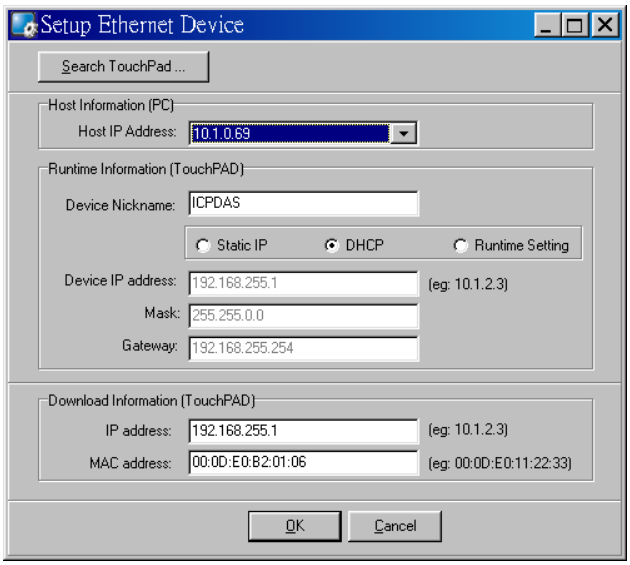
2. If the TouchPAD device is found and displayed in the list on the “**Search TouchPAD**” dialog box, double click on the TouchPAD item in the list to bring the information back to the “**Setup Ethernet Device**” window.

3. Select the IP Address type, “DHCP” or “Static IP”, for the new program and then press “**OK**”. Click the “**Run**” option from the “**Run**” menu to download the new program to the TouchPAD device. (We will introduce the details below.)

Important Notice

When using “Static IP” or “DHCP”, the IP settings are stored as a part of the program image, and only successful downloading can update the IP settings.

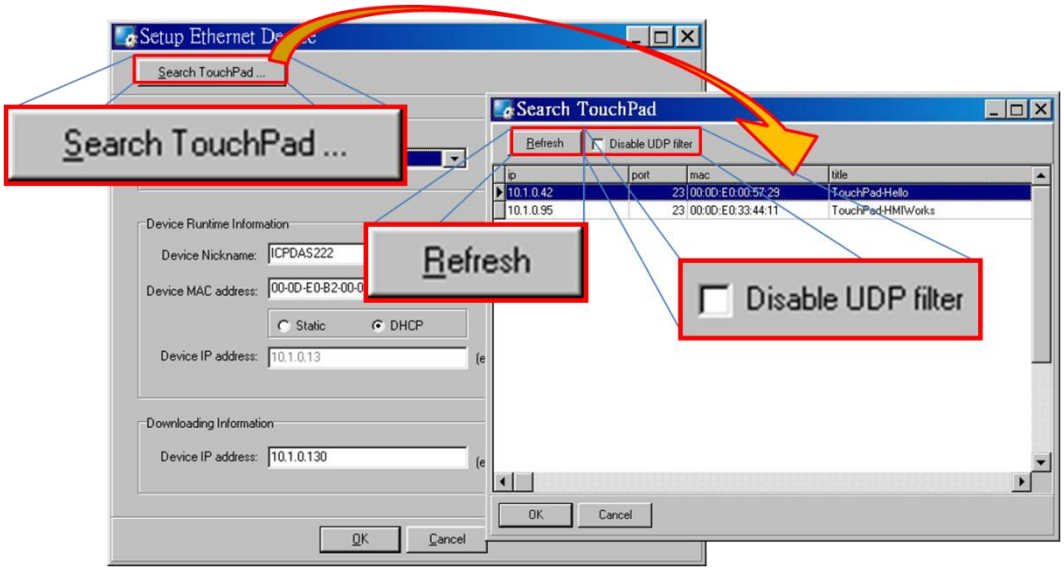
But if using “Runtime Setting”, the program image has no information about the IP settings, and TouchPAD loads the IP information from the flash at the runtime. Before IP settings are used, be sure to set the IP settings into the flash by the related API functions. We have demo to do this as well.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

item	description
Search TouchPAD	Search for the TouchPAD devices on the network. Make sure that the TouchPAD devices and the local computer are in the same subnet.
Host IP Address	The IP address of the local computer
Device Nickname	The nickname used to identify the TouchPAD device which is selected in the list on the "**Search TouchPAD**" window. This nickname is part of the program image and it takes effect after the new program runs (downloading successful).
Type of IP	Static IP or DHCP
Device IP Address (in the "Runtime Information" group)	The IP address of the TouchPAD device which is used in the runtime of the program. The IP address in the runtime is part of the program and it takes effect after the new program runs (downloading successful).
Mask	The subnet mask for the TouchPAD
Gateway	The gateway address for the TouchPAD. The gateway address is the IP address for a network interface on a router that leads to a larger network.
Device IP Address (in the "Download Information" group)	The IP address of the TouchPAD which is used only when downloading programs. This IP address information is NOT part of the program image, every time the downloading process starts, HMIWorks assigns this IP address just for downloading only.
MAC address	The MAC address of the TouchPAD device which is selected in the "**Search TouchPAD**" window. Every TouchPAD device is shipped with its MAC information pasted on its back panel.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



item	description
Refresh	Re-search again to re-make the list.
Disable UDP filter	We have UDP filter enabled by default to search for the TouchPAD devices only. Devices which are not TouchPAD are filtered out if this option is not checked.

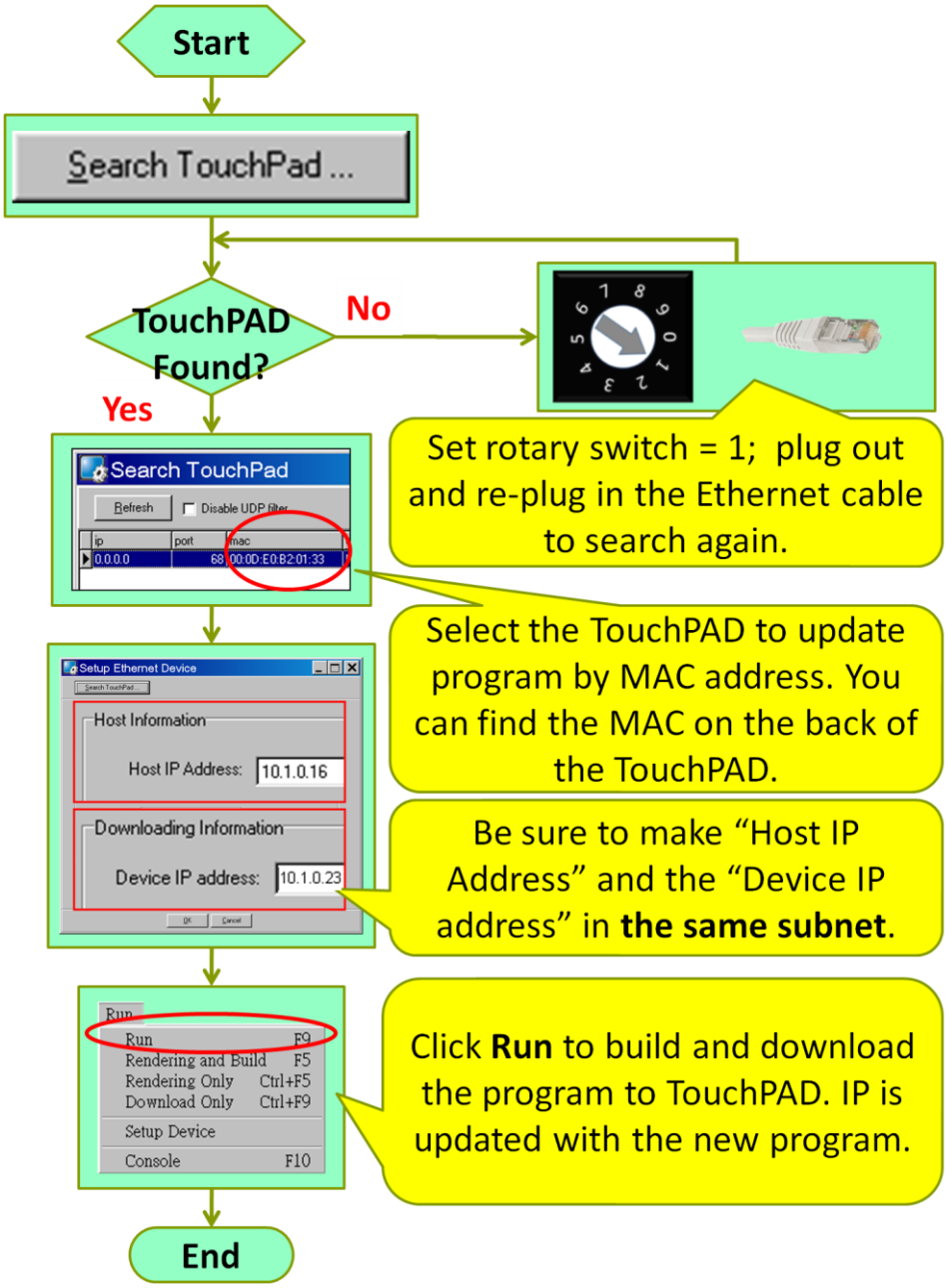
Press the **"Search TouchPAD"** button to search the TPD Ethernet device.

Note1: The MAC address can be found on the back panel of the TPD-283 device. HMIWorks uses this MAC address to search for the TouchPAD device. To see if the TouchPAD device is on the network, check that if there is a device with the MAC address in the list after searching.

Note2: Be sure to put the TPD Ethernet device and your PC in the same subnet.

In short, we draw the flow chart of setup the TPD Ethernet devices:

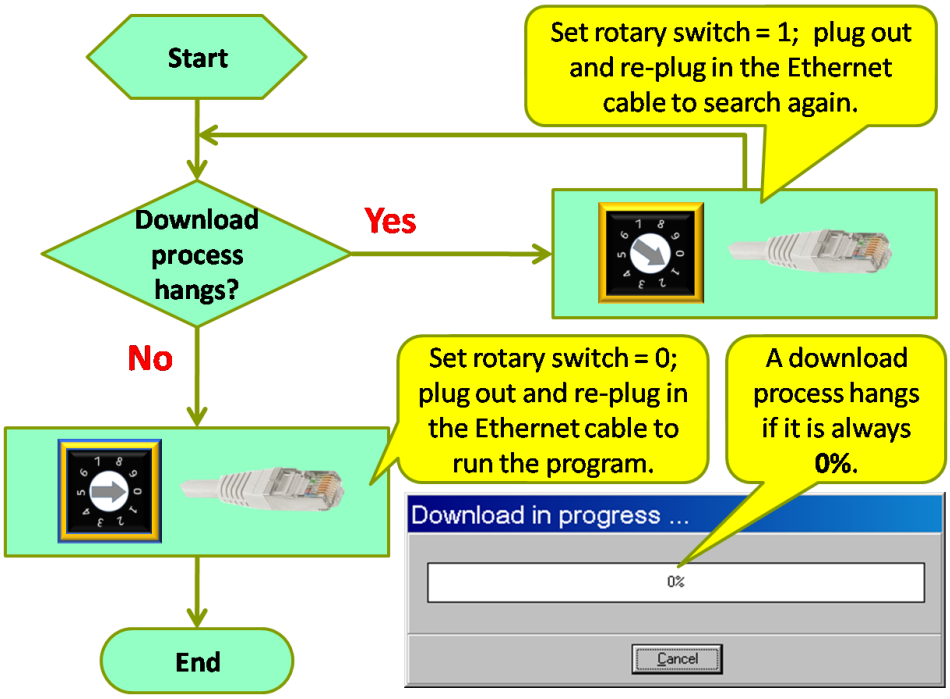
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

What to do if the download process hangs?

Anytime download process hangs, users can follow the flow below to complete the download process. (Note that below is not just for the case after setup device.)


Start
Set rotary switch = 1; plug out and re-plug in the Ethernet cable to search again.
Download process hangs?
Yes
No
Set rotary switch = 0; plug out and re-plug in the Ethernet cable to run the program.
A download process hangs if it is always 0%.
Download in progress ...
0%
Cancel
End

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4.2.3. Setup USB-downloaded devices (TPD-28xU, TPD-43x, TPD-43xF, VPD-13x, VPD-14x, VPD-133-H)

We use the rotary switch to set the configuration mode.

No	Mode	Description
0	Run	This mode is used to run the application. (There is only one application on a TouchPAD device.)
1	Update OS	This mode is used to update operating system of the TouchPAD device.
9	Update AP	This mode is used to download an application to the TouchPAD device. (There is only one application on a TouchPAD device.)

Unlike others series device, Users **need not** “Setup Device” from the “**Run**” menu because USB downloading requires no user-specified download information. But still some Ethernet settings may be **needed** in the “Setup Device” form for any Ethernet versions of TouchPAD.

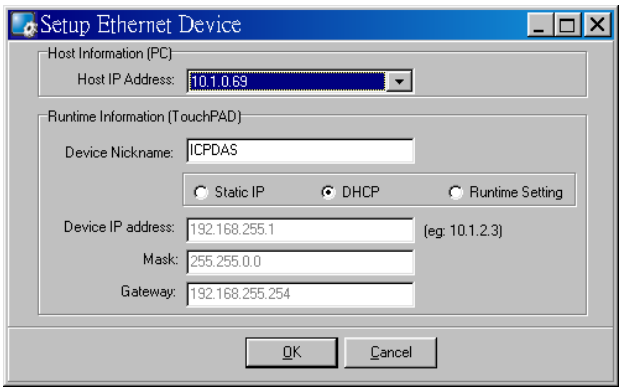
Notice

Other means of downloading (such as through the RS-485 network) are not provided for the TouchPAD devices which have a USB port on themselves.

Setup Ethernet Information for TouchPAD

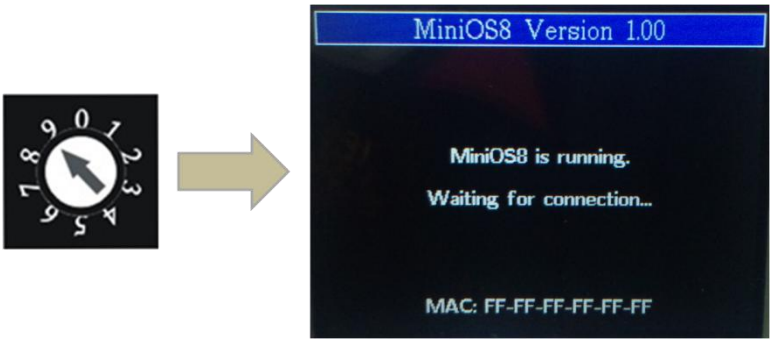
Setting up the Ethernet parameters for Ethernet versions of TouchPAD is the same as TPD-283 (whose program is downloaded by Ethernet). Since all Ethernet versions of TouchPAD except TPD-283 use USB to download their programs, refer to the previous section of TPD-283 about the Ethernet parameters but excludes the downloading information for more details.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Installing the TouchPAD USB Driver on your PC

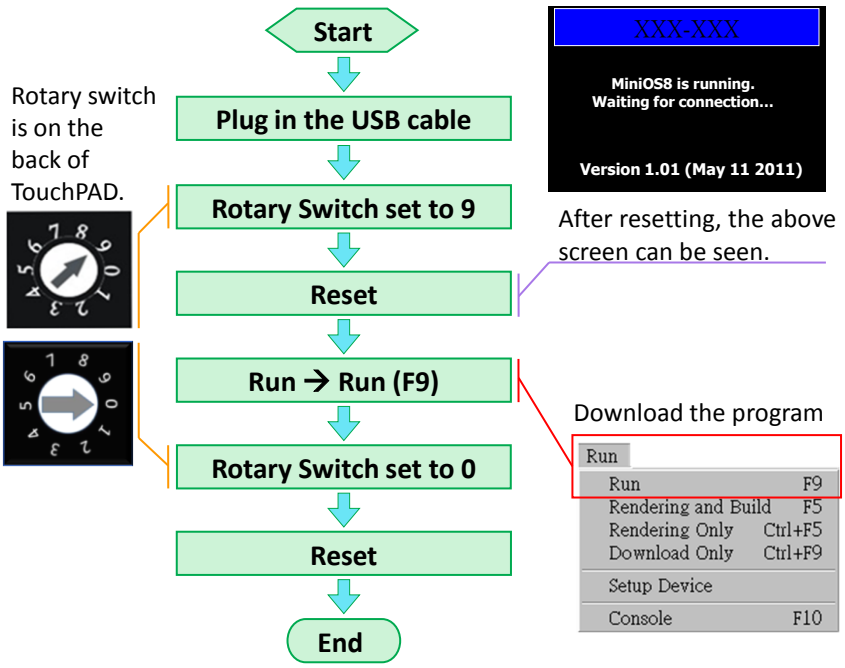
1. If HMIWorks version 2.03 or above has been already installed on your PC, the TouchPAD USB driver has also been automatically installed, too. However, users may need to update the TouchPAD USB driver manually if some previous version of HMIWorks was installed.
   HMIWorks setup file: ftp://ftp.icpdas.com/pub/cd/touchpad/setup/
2. To update the TouchPAD USB driver, use the USB driver located in the below directory to update the driver in the device manager from the control panel.
   C:\ICPDAS\HMIWorks_Standard\Tools\USB_Drivers
   ("C:\ICPDAS\HMIWorks_Standard\" is the installation path of the HMIWorks software.)
3. In order to update a USB driver for the TouchPAD devices, set the rotary switch on the TouchPAD device to the "9" position, then plug the USB cable into the TouchPAD Device and then turn on the supply power to the TouchPAD device (reset). Finally, you can see that the TouchPAD USB driver is in the list of the device manager. (something like "Stellaris Device Firmware Upgrade")



Downloading programs through USB

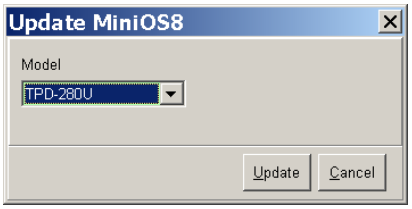
Follow the flow to download a program to the TouchPAD device.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Updating OS through USB

Some devices (e.g. TPD-280) in the TouchPAD series do not have MiniOS8 on them, but still some do have. To update MiniOS8, go to the **File** menu and click the “Update MiniOS8” option to open the dialog as shown below.



The Model drop down list shows the models in the TouchPAD series that have MiniOS8. Select the model of the TouchPAD to update.
Note: each version of HMIWorks corresponds to a version of MiniOS8。Though we try to achieve backward compability, we still suggest update MiniOS8 every time HMIWorks is updated.

Application Image Checksum

Using MiniOS8.bin of version 1.1.8 (corresponding HMIWorks version 2.07) or above, TouchPAD supports application image checksum when loading the application program from the flash at startup.
If checksum error when loading the application program, the below message is

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

shown on the screen, (Application Checksum Error!).

Try to download the application program again to solve this error.

Set this option at:
HMI -> “Project Configuration” -> “General 2” -> “Application Image Checksum”

4.2.4. Calibrations

Usually users need not to calibrate the touch screen because we calibrate the TouchPAD devices before shipping. However, in cases users may need to calibrate the touch screens, we introduce the flow below.

You can find that the calibration programs are in the following directory: “[HMIWorks_Install_Path]\bin\boot\[Device_Name]\calibrate”. For example, there are two calibration programs can be found at “C:\ICPDAS\HMIWorks_Standard\bin\boot\TPD-280U\calibrate” if trying to calibrate TPD-280U. One is for landscape (when the screen is horizontal) and the other is for portrait (when the screen is upright). And in the same directory of the file, calibrate.bin, there’s a batch file which is used to download the calibrate.bin to the TPD-280U device and it is called “calibrate.bat”.

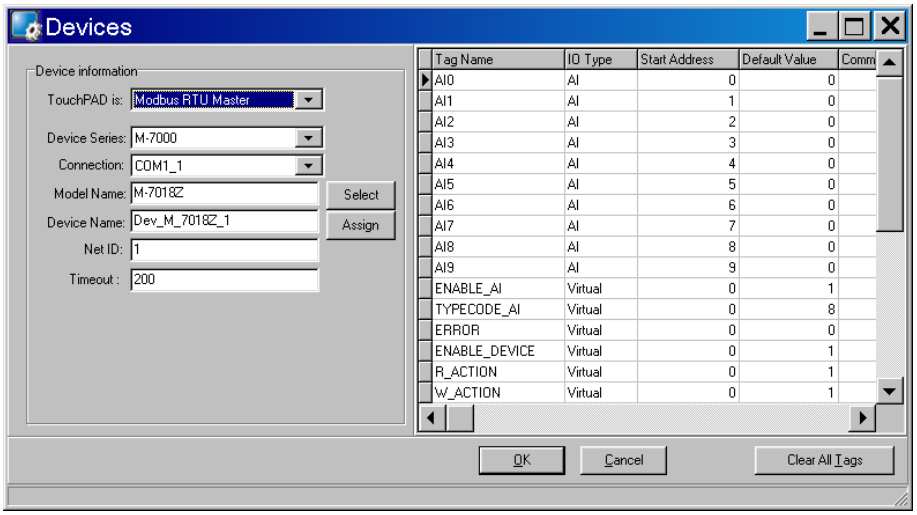
Follow the flow to calibrate the TouchPAD device.

4.3. Connecting to I/O Devices

We provide connection methods for three series of I/O modules, the PET-7000, the I-7000, and the M-7000 series and a general approaches for the Modbus TCP Master I/O modules, Modbus RTU Master/Slave I/O modules.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Click the “**Register Devices (I/O)**” option from the “**HMI**” menu or press the “**F3**” on your keyboard and then the “**Devices**” window is displayed.



Step by step specify or fill each field and click the **OK** button to import tags. Finally, check these imported tags in the **Workspace**.

The possible device series are as below:

TouchPAD is	Device Series	Device Series Description
Modbus RTU Master	M-7000	Remote I/O modules over Modbus RTU protocol
	XVBoard	VPD series I/O Expansion Boards
	User_Define (MRTUM)	Remote Modbus RTU I/O modules of third parties
Modbus RTU Slave	Profiles (MRTUS)	TouchPAD is treated as a slave device and wait for some master devices to control
DCON Master	I-7000	Remote I/O modules over DCON protocol
Modbus TCP Master	PET-7000	Remote I/O modules over Modbus TCP protocol
	WISE-7000	WISE (Web Inside, Smart Engine) devices
	User_Define (MTCPM)	Remote Modbus TCP I/O modules of third parties
Modbus TCP Slave	Profiles (MTCPS)	TouchPAD is treated as a slave device and wait for some master devices to control

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

The explanations of items in the **Devices** window (Register Device):

Item	Description
Connection	Specify an existing connection approach (TCPIP, XVboard or UART) or create a new one to connect to the I/O module. Note: when TouchPAD is Modbus TCP slave, check the **As a Server** option in the **New/Edit Connection** dialog which is called by right click on the connection item in the **Workspace**.
Device Name	Specify the name of the I/O module. Users can assign a name they want.
Model Name	Specify the model name of the I/O module to connect.
Net ID	When TouchPAD acts as a master device, Net ID is the specified ID of the I/O module in the network. Possible range: For Modbus RTU: 1 ~ 247 For DCON: 0 ~ 255 For PET-7000: 1 ~ 255
	When TouchPAD acts as a slave device, Net ID is the specified ID of TouchPAD itself.
Timeout	The timeout value for the communications, both RS-485 and TCP

4.4 Connecting to I/O Expansion Boards (XVBoard)

Default value

Net ID	1
Baud Rate	115200
Comport	XVBus
Communication interface	Modbus RTU(XVBoard as Slave)

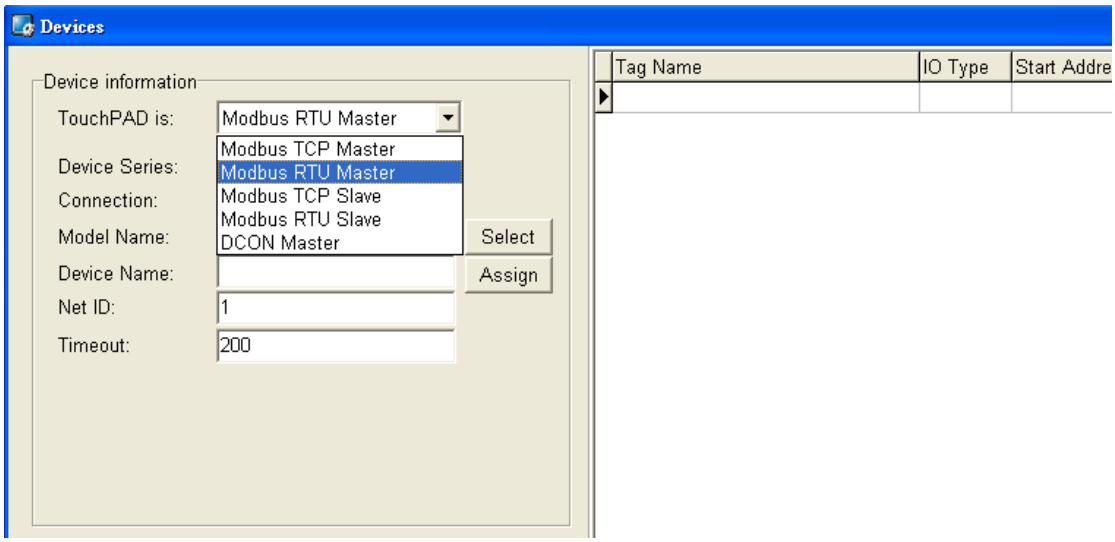
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Using C programming language to control XVBoard

```
HANDLE h;
int NetID = 1;
int addr = 1;
int ch_count = 8;
char DO_value[1];

DO_value[0] = 3;    //that is, turn on the ch 0 and ch1.
h = uart_Open("XVBus,115200,N,8,1");
mrm_WriteDO (h, NetID, addr, ch_count, DO_value);
uart_Close(h);
```

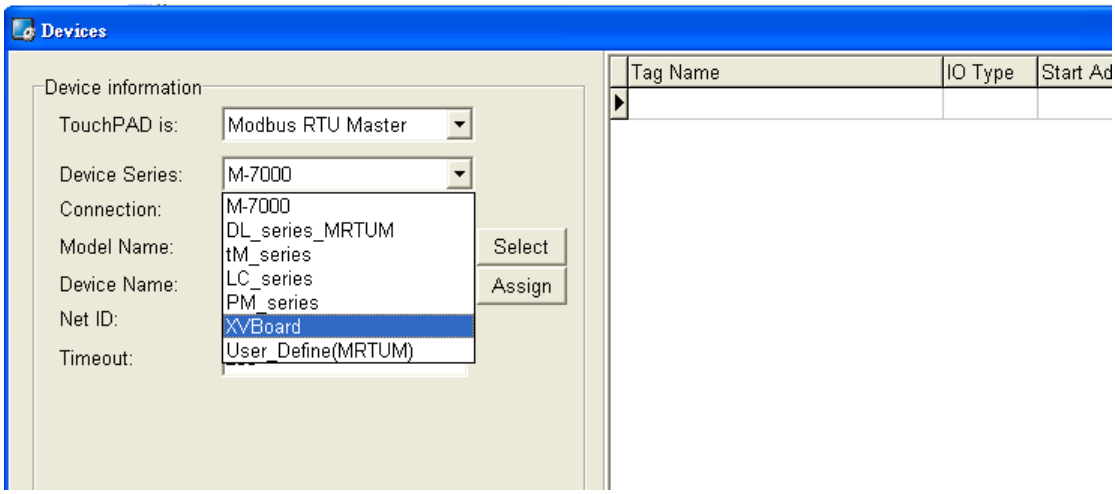
Using Ladder designer

1.After adding a new device, select **TouchPAD is** as **Modbus RTU Master**.

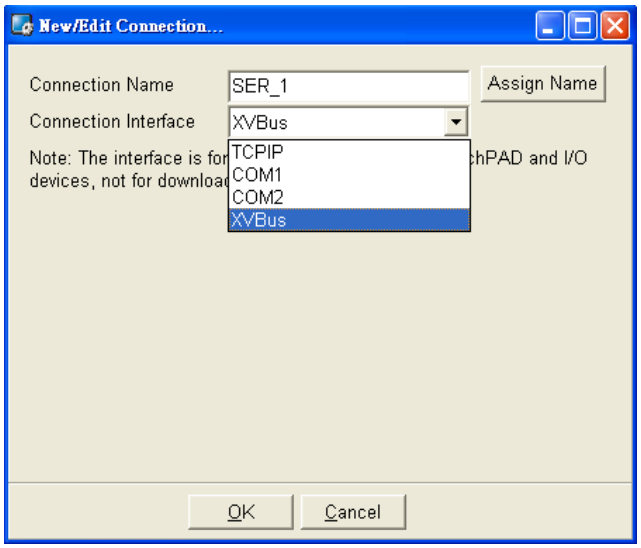


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

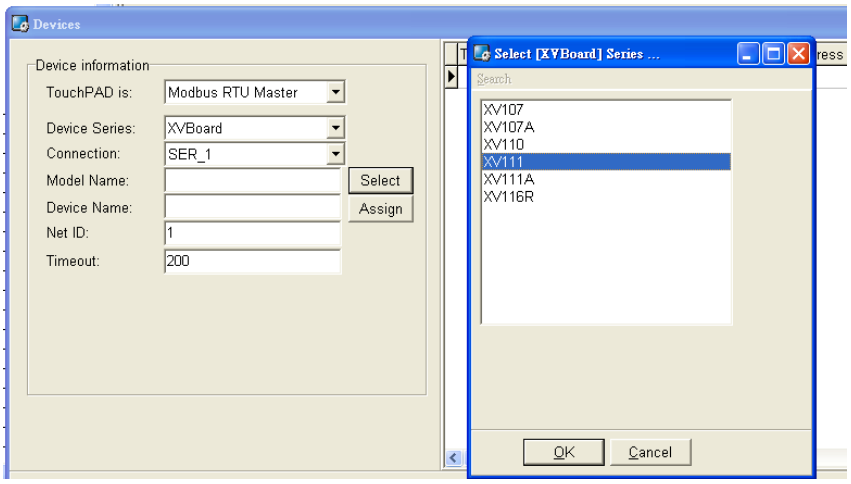
2. Select **Device Series** is **XVBoard**.



3.Edit the connection, select **Connection Interface** is **XVBus**



4.Press the **Select** button, and then select the model.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5. Development Software, HMIWorks

HMIWorks is the development tool for, both TPD and VPD series. It supports two programming types, Standard C and Ladder. Compared with traditional GUI development tools, HMIWorks is easy to learn, flexible to design GUIs, and takes less time to raise productivity.

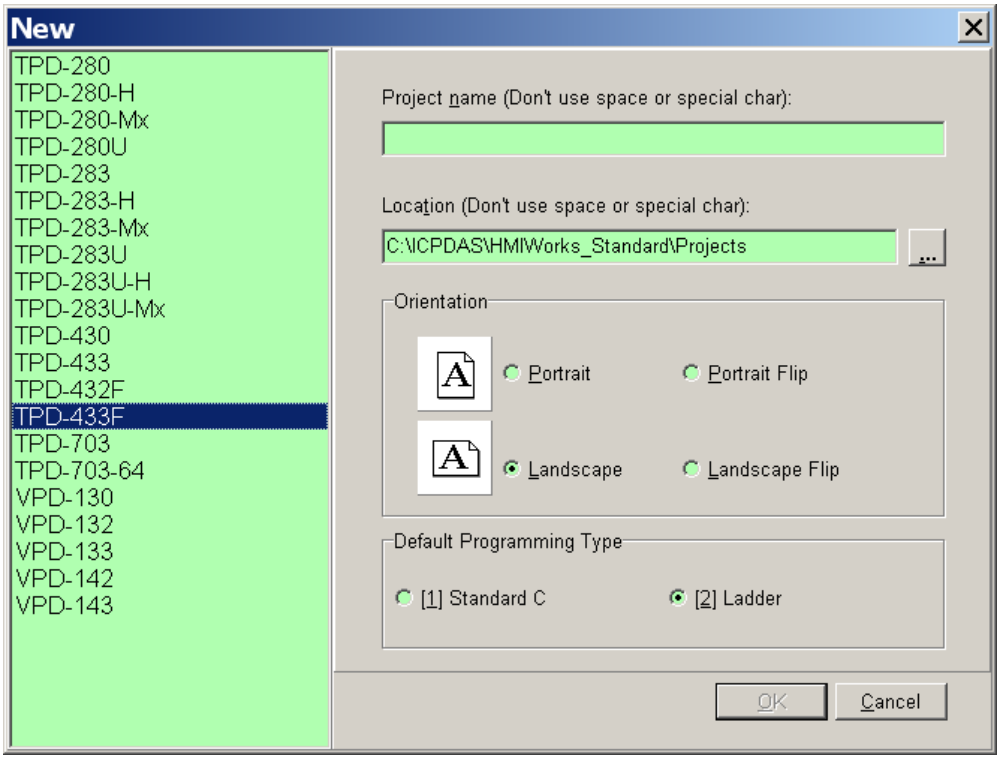
Features of HMIWorks include:

- FREE of charge (for ICP DAS TouchPAD devices)
- Two programming types, ladder diagram and Standard C
- Plenty of widgets
- Plenty of demos shorten development time
- Advanced search for I/O modules
- Detail error messages
- Easy downloading after building
- Automatic generated codes for user-designed frames
- Multi-frame design
- Abstract graphics as simple APIs
- Easy learning IDE to raise productivity in short time
- Data exchange function

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.1. The Construction of HMIWorks

Before showing the construction of HMIWorks, create a new project first. Click **File** menu, then click on **New…**.



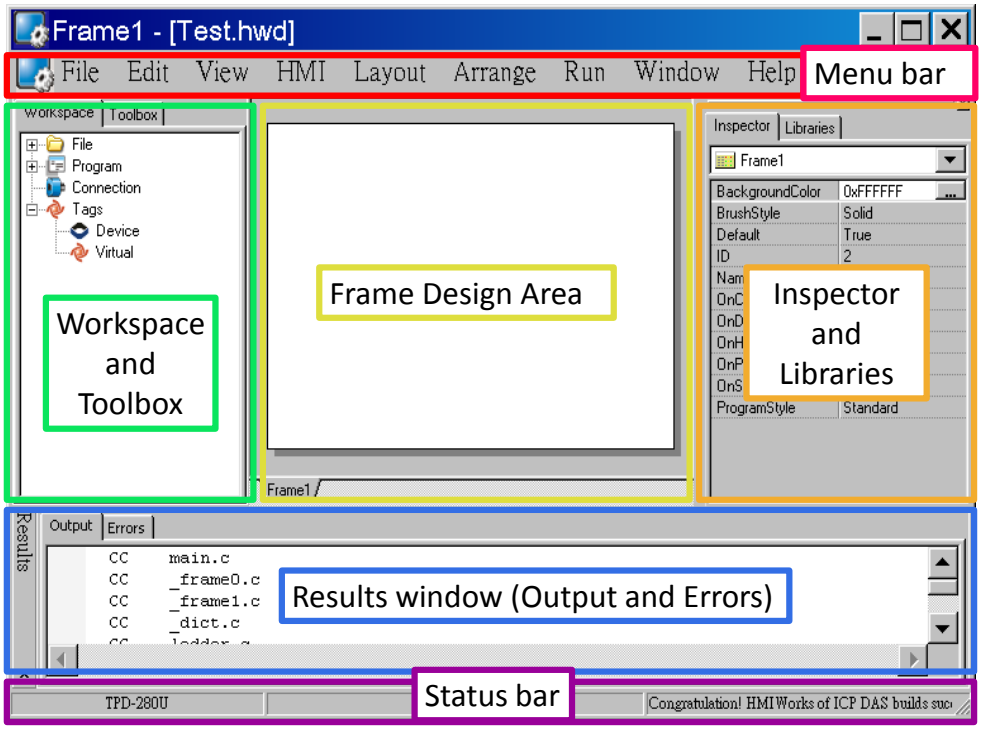
Notice

A *valid project name* is a sequence of one or more letters, digits or underscore characters (_). It must not begin with a digit. Besides, it is of suggested length 100 characters (including its path).

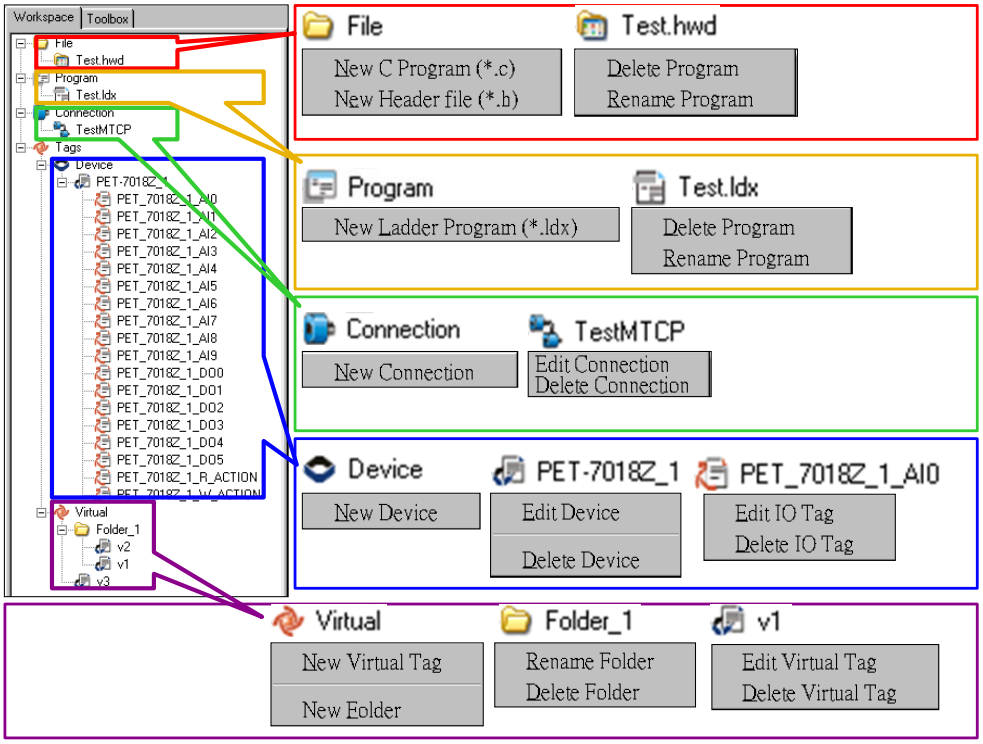
Choose the target module, **Orientation**, and the **Default Programming Type**. Press **OK** and HMIWorks integrated design environment shows as below. There are several parts of HMIWorks.

1. Menu bar
2. Workspace and Toolbox
3. Frame Design area
4. Inspector and Libraries
5. Results window (Output and Errors)
6. Status bar

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Below are the operations (pop-up menus) that users have in **Workspace**.



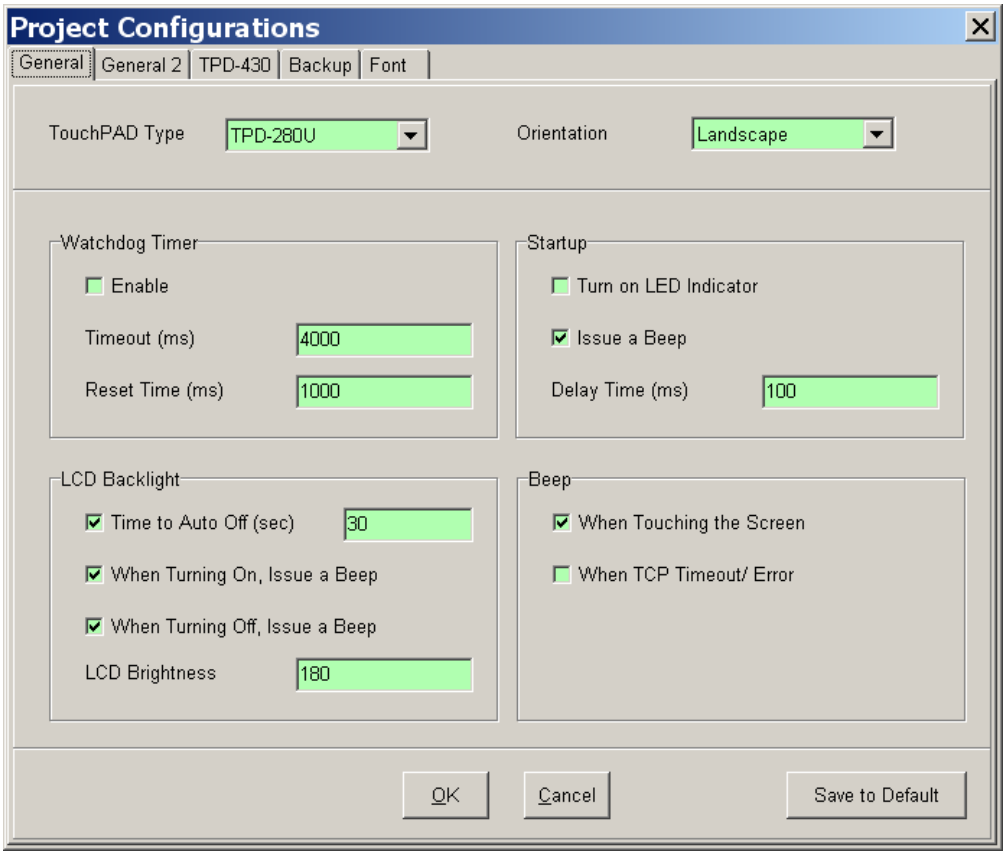
Next sections show the functions of these parts.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.2. The Options of TouchPAD

5.2.1 Project configurations

HMI -> Project configurations



Some important options

Tab	Option	Description
General	**TouchPAD Type & Orientation**	After changing these two options, HMIWorks automatically scale the size of every frame and every widget to maintain the relative positions between each other. **Note**: the Text component is not scaled.
	LCD Brightness	Range: 0 ~ 255. 0: the darkest 255: the brightest
	Beep	If “When Touching the Screen”

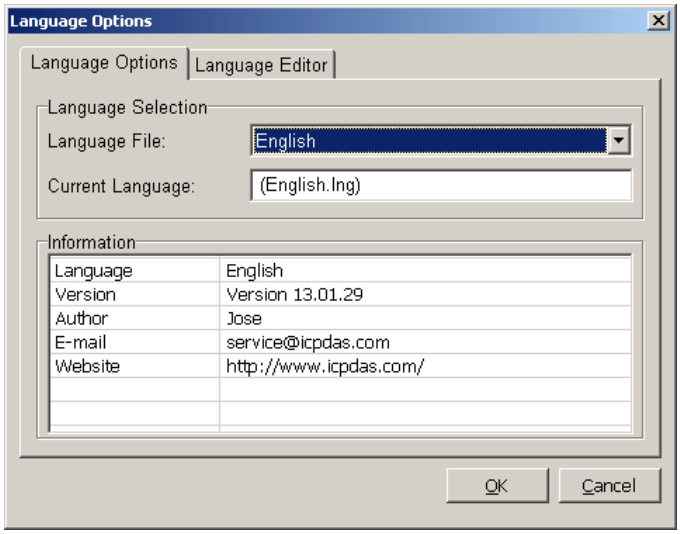
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

		checked, the hmi_PlaySong function becomes useless.
General 2	**Application Image Checksum**	Calculate the checksum when loading the application program. See the section, "Setup Other Devices in TouchPAD Series" for more information.
	Refresh Time	The period of both I/O scan and touch screen refresh.
	Connecting Blinking Cycle	Used for communications of Modbus TCP master polling (remote slave devices), the Connecting Blinking Cycle defines the blinking period of "ERROR" tag used in devices which can be found in the Workspace.
	Reconnection Interval	The interval between two groups of 7 consecutive connection tries.
Backup	**Backup when a project is closed**	HMIWorks can backup when a project is closed. The backup files are compressed in the format, .7z.
Font	**Language support**	Besides English(ASCII, 0x20--0x7E), now we support Russian (U+0410--U+044F). Note: when using Russian, English(0x20--0x7E) is also supported.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.2.2 Language Options

View -> Language Options.

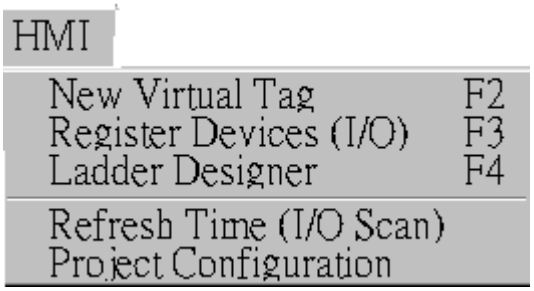


5.3. Ladder Designer

One of the most important features of HMIWorks is Ladder Designer.
The ladder logic is defined by the followings:

1. A Ladder Diagram consists of many rungs.
2. Each rung resembles a circuit which is formed by relays.
3. All of the rungs are executed serially in a loop.

Click **HMI** menu to use this feature.



- **New Virtual Tag**: defines your own variables
- **Register Devices (I/O)**: uses I/O devices of ICP DAS on the networks
- **Ladder Designer**: designs your ladder logics

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

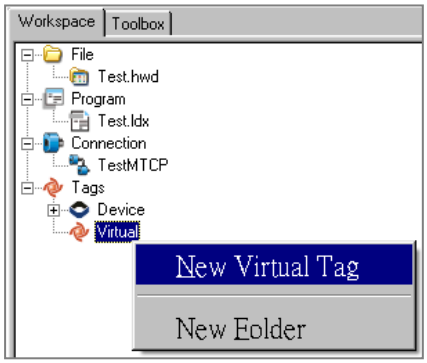
- **Refresh Time (I/O Scan)**: set the refresh time of each scan of a Ladder (the minimum value is 100 ms) → depreciated, this item is moved to the "Project Configuration".
- **Project Configuration**: the configuration of the project

Users can manage their ladder design in the **Workspace**.

5.3.1. Getting Started

- To use the **Ladder Designer**, run HMIWorks_Standard.exe to create a new project first.

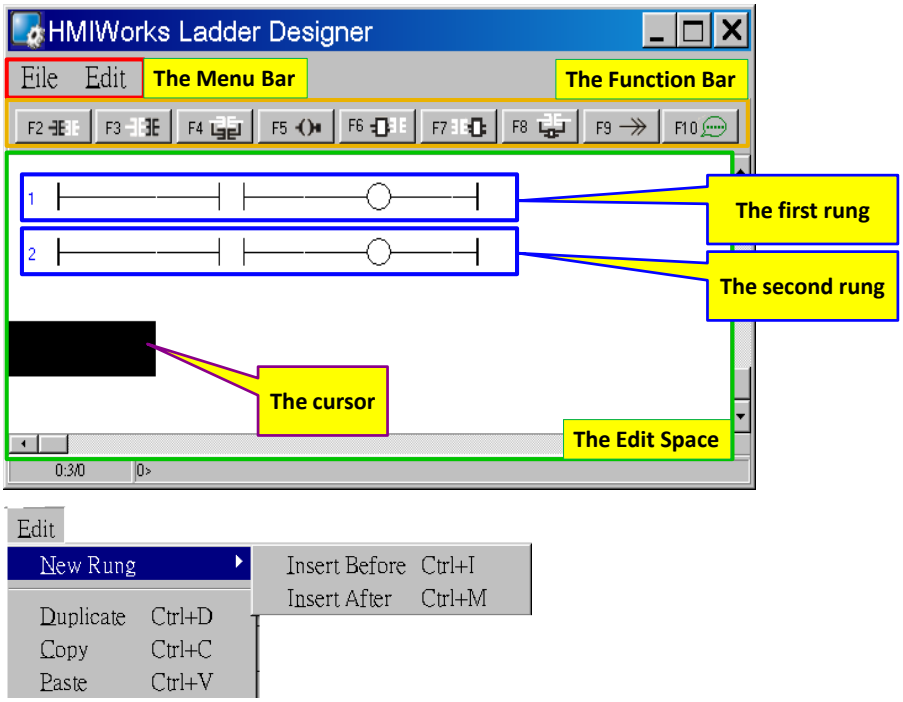
- **New Virtual Tag** and open **Ladder Designer** from the **HMI** menu
- **New Virtual Tag** adds variables used in the **Ladder Designer**.
  There are three ways to open the "**Edit variable**" window,
  - pressing **F2** key on your keyboard,
  - select the **New Virtual Tag** option in the **HMI** menu,
  - right-click on the **Virtual** item and click the "**New Virtual Tag**" option.



5.3.2. Introduction to Ladder Designer

A **Ladder Designer** is a tool to implement the ladder logic according to users' design.
Press **F4** on your keyboard to open the **Ladder Designer**.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Mainly, a **Ladder Designer** consists of three parts, the menu bar, the function bar, and the edit space. The highlighted rectangle area is the cursor.

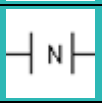
The briefings of the function bar:

Item	Description
F2	Insert a contact input in the left of the cursor
F3	Insert a contact input in the right of the cursor
F4	Insert a contact input which is parallel to the cursor
F5	Insert a coil output
F6	Insert a function block in the left of the cursor
F7	Insert a function block in the right of the cursor
F8	Insert a function block which is parallel to the cursor

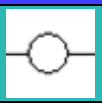
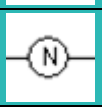
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

F9	Insert a Jump which is parallel to the cursor
Space [T]	~~Change the type of the contact input/ coil output~~
F10	Add comments

The briefings of the contact input type:

Item	Description
	A normally-open contact input
	A normally-closed contact input
	A positive transition contact input ➢ when the state from OFF to ON, trigger one shot
	A negative transition contact input ➢ when the state from ON to OFF, trigger one shot

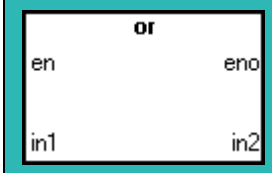
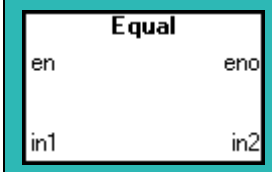
The briefings of the coil output type:

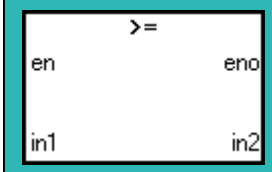
Item	Description
	A normally-open coil output
	A normally-closed coil output
	A “Set” coil output ➢ once triggered, the coil remains ON until a reset
	A “Reset” coil output ➢ once triggered, the coil remains OFF until a set
	A positive transition coil output ➢ when the state from OFF to ON, trigger one shot
	A negative transition coil output ➢ when the state from ON to OFF, trigger one shot

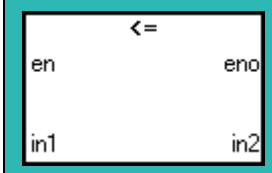
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

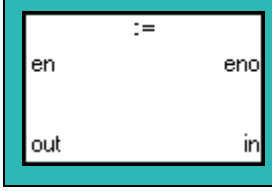
The briefings of function blocks

Refer to “C:\ICPDAS\HMIWorks_Standard\bin\FunctionBlock” for more details.

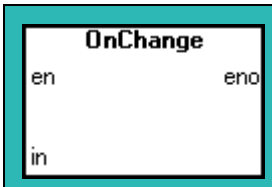
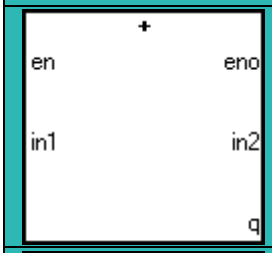
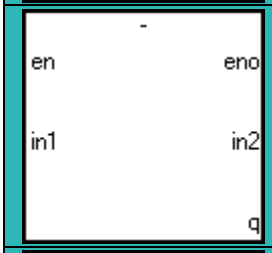
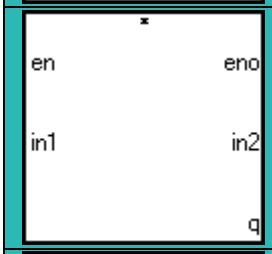
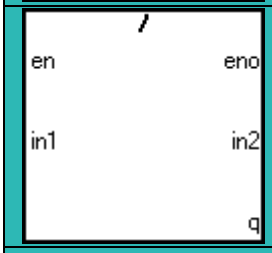
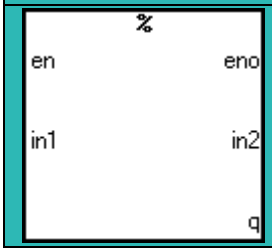
Item	Description (parts of C code)	Group
	AND (And) If en == 1, eno = in1 & in2; Else eno = 0.	default
	**OR (Or)** If en == 1, eno = in1 | in2; Else eno = 0.	default
	XOR (Exclusive Or) If en == 1, eno = in1 ^ in2; Else eno = 0.	default
	**Equal** If (en == 1 and in1 is equal to in2), eno = 1; Else eno = 0;	default
	NE (Not Equal) If (en == 1 and in1 is not equal to in2), eno = 1; Else eno = 0;	default

	**GE (Greater or Equal)**
	If (en == 1 and in1 >= in2),
	eno = 1;
	Else eno = 0;
		default

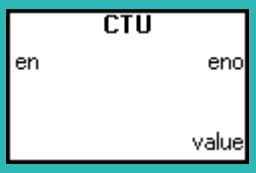
	**LE (Less or Equal)**
	If (en == 1 and in1 <= in2),
	eno = 1;
	Else eno = 0;
		default

	**Assign** If en == 1, “out” is assigned with “in” and eno = 1; Else eno = 0;	default

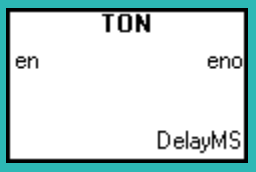
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

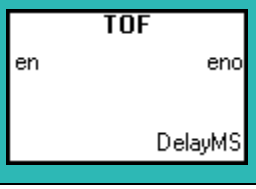
	**OnChange** If en == 1 and “in” is changed, eno = in; Else eno = 0;	default
	**Add (Addition)** If en == 1, q = in1 + in2 and eno = 1; Else eno = 0;	math
	**Sub (Subtraction)** If en == 1, q = in1 - in2 and eno = 1; Else eno = 0;	math
	**Mul (Multiplication)** If en == 1, q = in1 * in2 and eno = 1; Else eno = 0;	math
	**Div (Division)** If en == 1, q = in1 / in2 and eno = 1; Else eno = 0;	math
	Inc (Increment) If en == 1, increment “in” by 1; Else eno = 0;	math
	Dec (decrement) If en == 1, decrement “in” by 1; Else eno = 0;	math
	**Mod (Modulo)** If en == 1, q = in1 % in2 and eno = 1; Else eno = 0;	math

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

	**CTU (Count Up)**
	End: count >= value.
	If en == 1,
	Count up until End,
	During counting, eno = 0,
	When End, eno = 1;
	Else
	Reset count to 0,
	eno = 0;
	
	Note: the counting period depends on the number of rungs
		counter

	CTD (Count Down)
	End: Count <= 0.
	If en == 1,
	Count down until End,
	During counting, eno = 0,
	When End, eno = 1;
	Else
	Reset count to value,
	eno = 0;
	
	Note: the counting period depends on the number of rungs
		counter

	**TON (Timer On, unit=ms)**
	End: elapsed >= DelayMS.
	If en == 1,
	Start the timer if not,
	Stop the timer when End,
	When timer runs, eno = 0,
	When End, eno = 1;
	Else
	Reset the timer,
	eno = 0;
		timer

	**TOF (Timer Off, unit=ms)** End: elapsed >= DelayMS. If en == 1, Start the timer if not,	timer

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

	Stop the timer when End,
	When timer runs, eno = 1,
	When End, eno = 0;
	Else
	Reset the timer,
	eno = 0;	

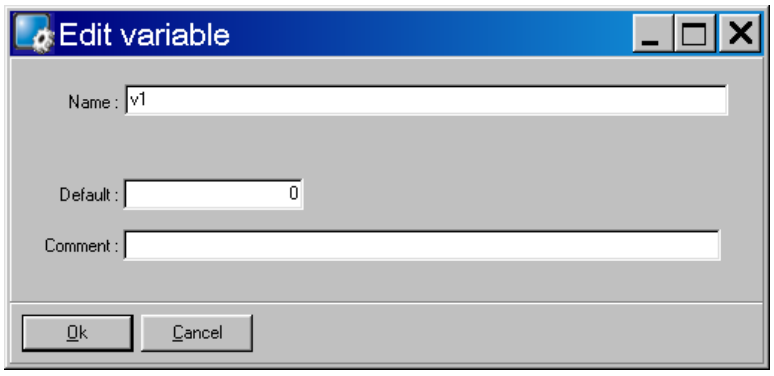
Beep en eno	**Beep** If en == 1, beep and eno = 1; Else eno = 0;	system

5.3.3. Operations of Ladder Designer

5.3.3.1. New Virtual Tags (F2)

To use the **Ladder Designer**, add variables for the **Ladder Designer** first.

1. Press **F2** on your keyboard or click the "**New Virtual Tag**" option from the "**HMI**" menu to add virtual tags, then an "**Edit variable**" window displayed.
2. Define a new variable in the "Name" field and optionally fill the other fields.
3. Finally, press the **OK** button to take effect.

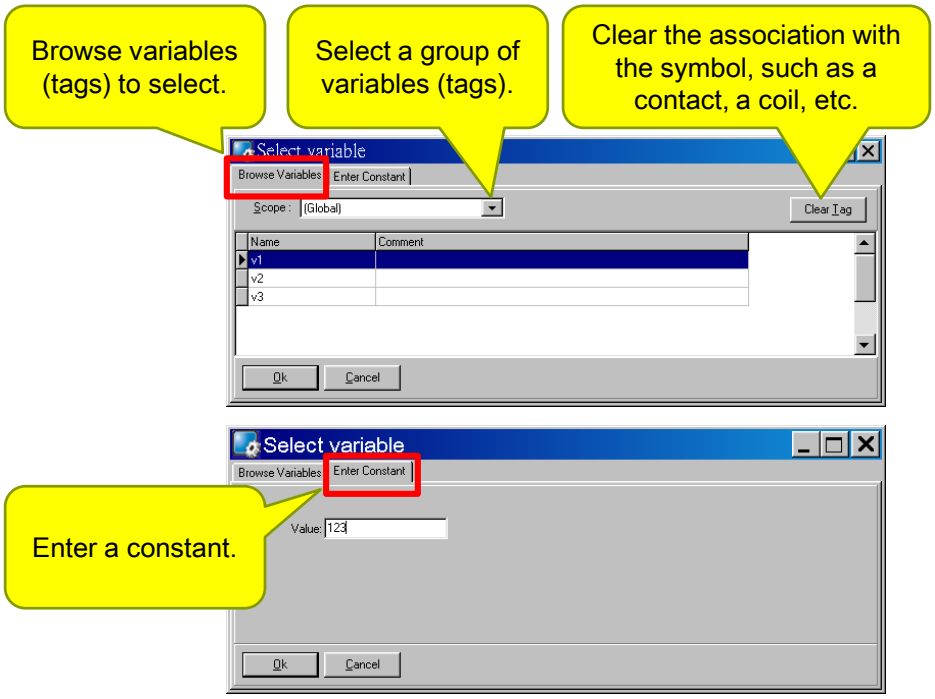


Here, we add three variable v1, v2 and v3 for example in the next sections.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

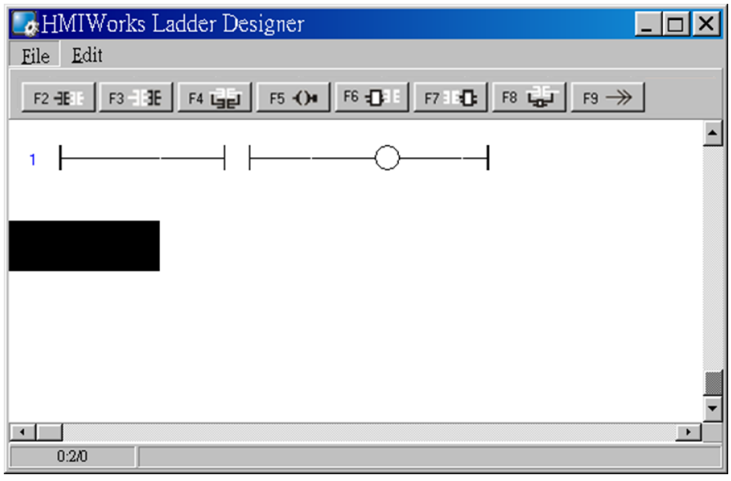
5.3.3.2. Assigning Variables and Constants

Double click on the symbol of contact inputs, coil outputs, etc. to open the "**Select variable**" window to select variables or enter constants as below.



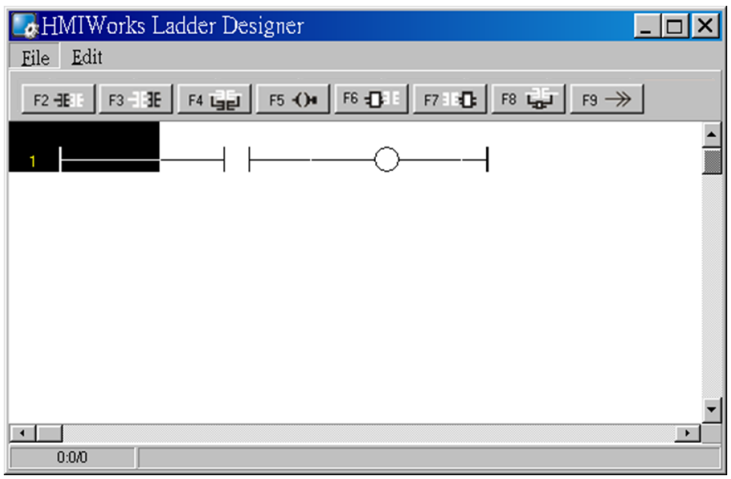
5.3.3.3. Inserting and Deleting a Rung

To insert a rung, move the cursor (the highlighted area) to the empty place and then press **F2** (or **F3**/**F4**) on your keyboard.
(Or press **F6**, **F7**, **F8** to insert a rung with a function block.)



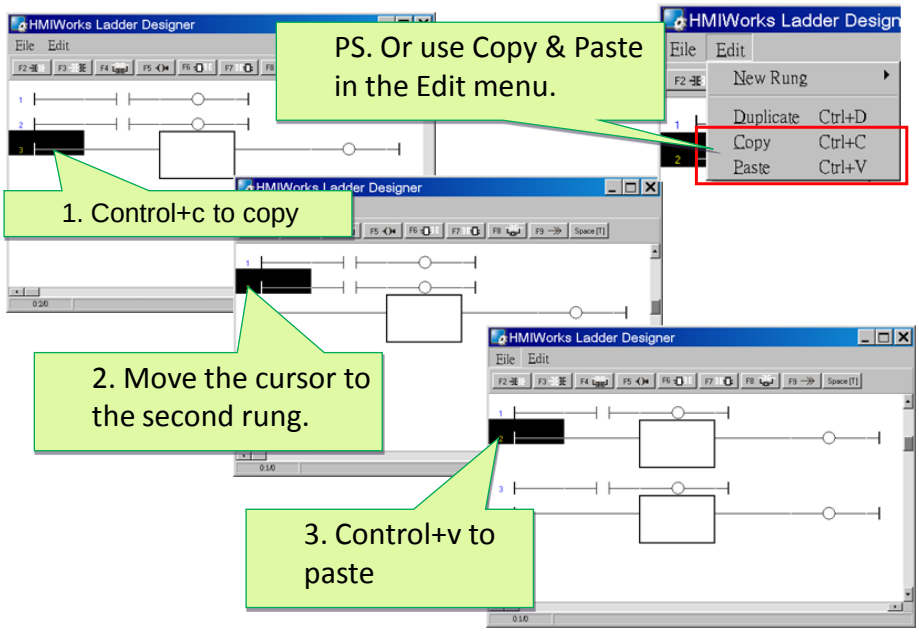
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

To delete a rung, move the cursor to the starting point of the rung and then press “**Delete**” key.



5.3.3.4. Copying and Pasting a Rung

Supposed that we have three rungs and we want to copy the third rung and insert it between the first and the second rungs.



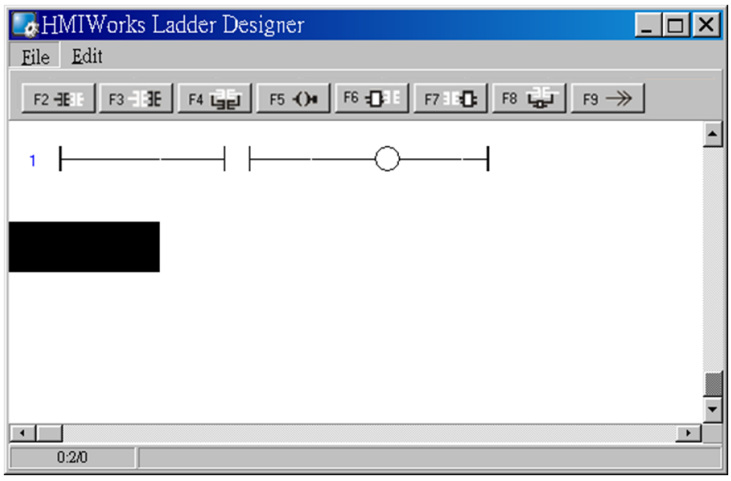
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.3.3.5. Inserting and Deleting a Contact Input

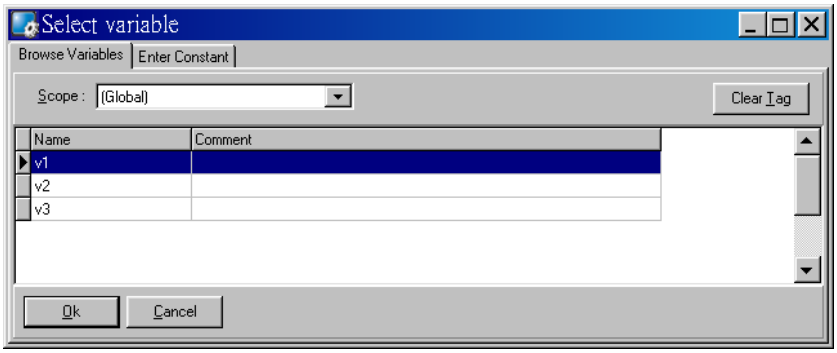
To demonstrate how to insert or delete a contact input and other related issues, go through the steps below.

1. Associate a variable to a contact input

Press **F2** on your keyboard to insert a new rung with a contact input and a coil output.

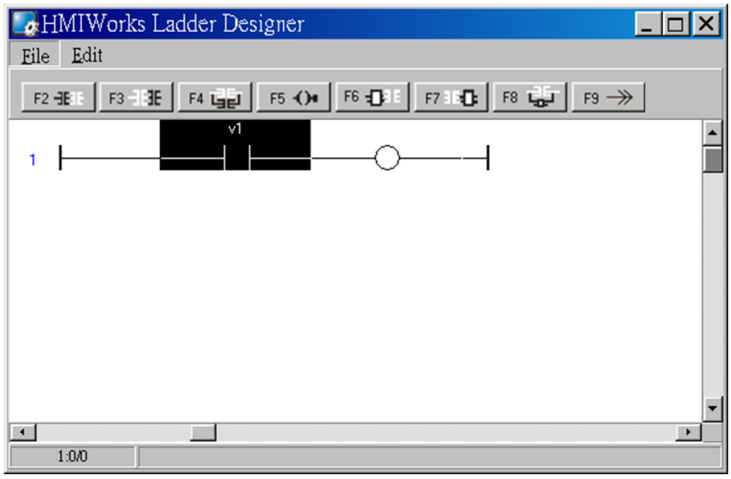


In the new rung, double-click on the contact input to open the "**Select variable**" window to select a variable (tag) and assign it to the contact input.



For example, we double-click on the variable "v1" and set to the contact input. v1, v2 and v3 are the variables set by "**New Virtual Tags**". Refer to the "**New Virtual Tags**" section.

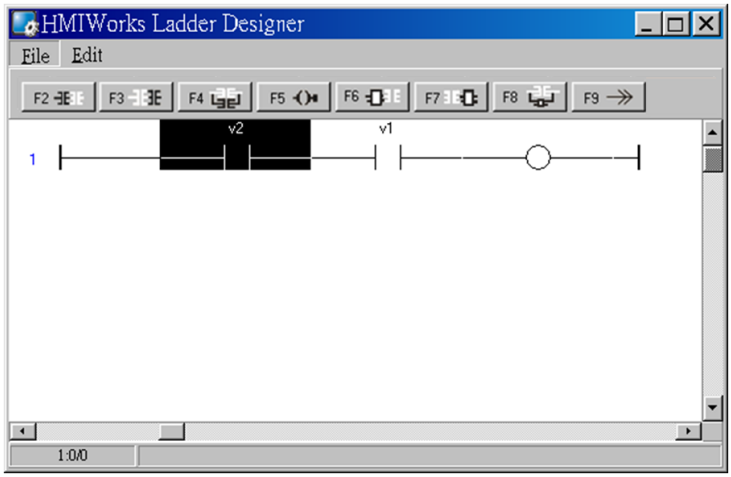
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



2. Insert a new contact input in the left of the cursor (**F2**)

Move the cursor to the “v1” contact input and then press **F2**.

And to make things clear, associate variable “v2” to the newly-inserted contact input.

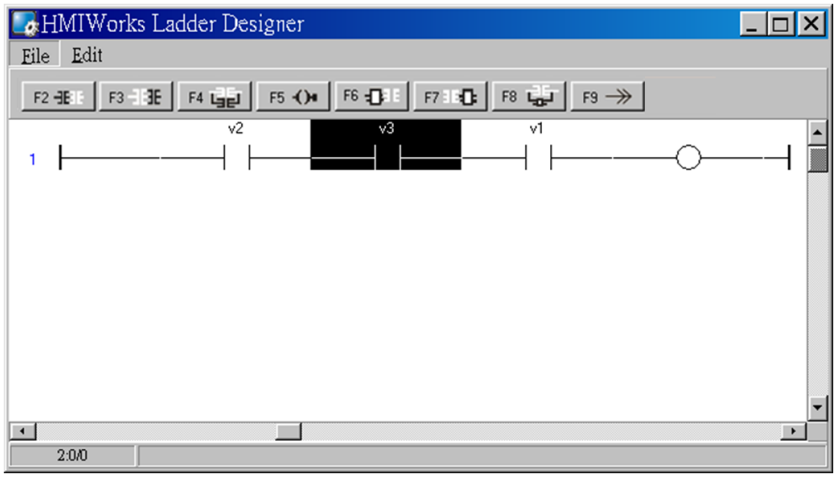


3. Insert a new contact input in the right of the cursor (**F3**)

Move the cursor to the “v2” contact input and then press **F3**.

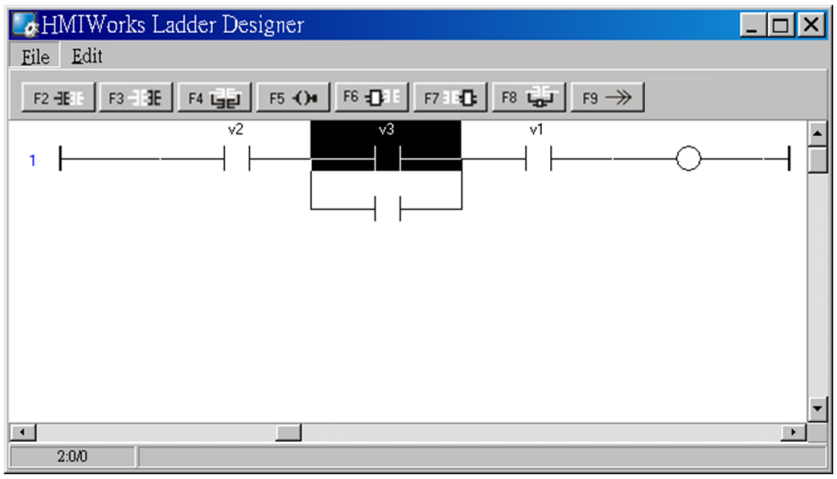
Associate variable “v3” to the newly-inserted contact input.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



4. Insert a new contact input which is parallel to the cursor (**F4**)

Move the cursor to the “v3” contact input and then press **F4**.

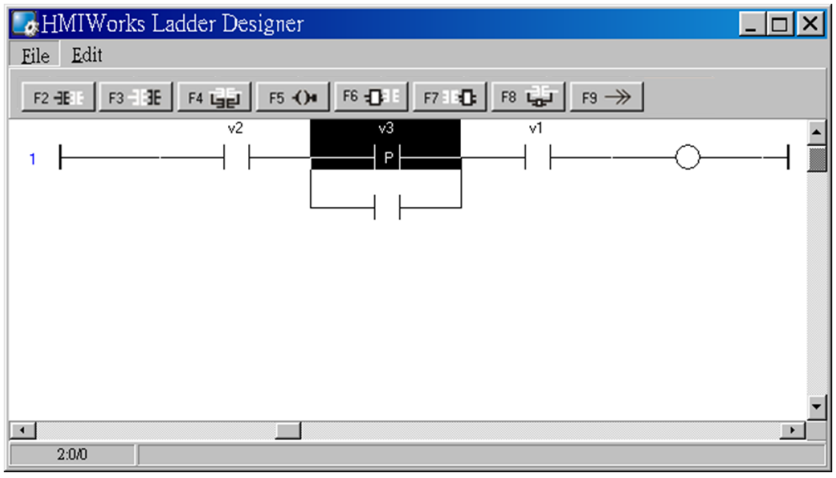


5. Set the type of a contact input

Move the cursor to a contact input and then press the space bar to change the type of the contact input.

For example, we move the cursor to the “v3” contact input. Press the spacebar twice to set the type of the contact input to pulse contact input.

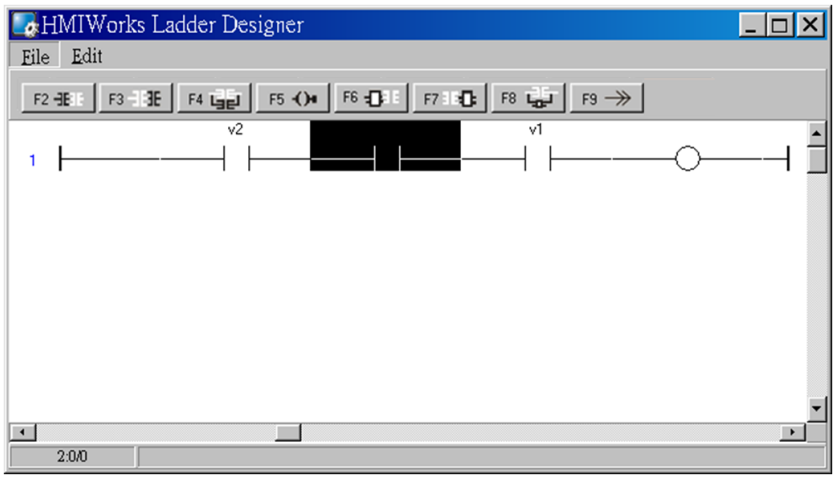
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



6. Delete a contact input in the rung

Move the cursor to the contact input you want to delete. Then press “**delete**” on your keyboard.

For example, we move the cursor to the “v3” contact input and then press the “**delete**” key.



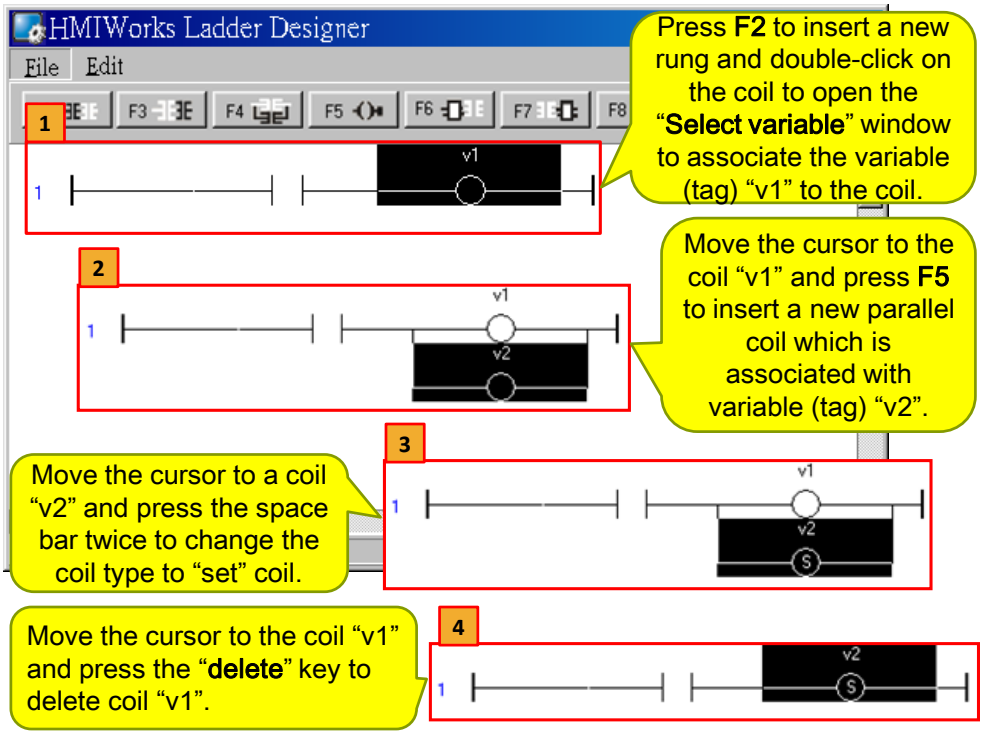
7. Delete the rung.

Move the cursor to the starting point of the rung and then press “**Delete**” on your keyboard.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.3.3.6. Inserting and Deleting a Coil Output

To demonstrate how to insert or delete a coil output and other related issues, see the figure below.

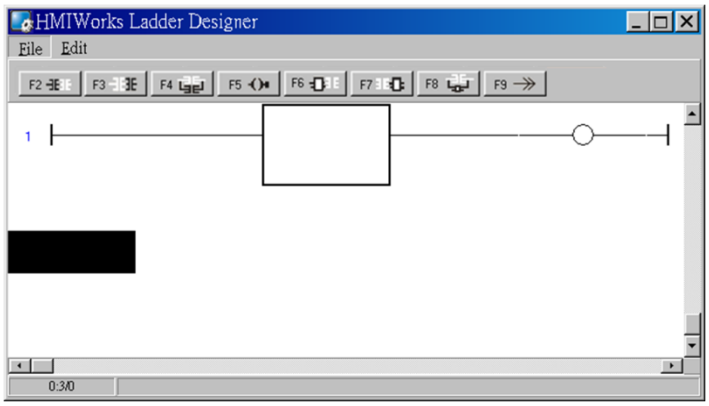

HMIWorks Ladder Designer
File
Edit
F3
F4
F5
F6
F7
F8
1
v1
Press F2 to insert a new rung and double-click on the coil to open the "Select variable" window to associate the variable (tag) "v1" to the coil.
2
v1
v2
Move the cursor to the coil "v1" and press F5 to insert a new parallel coil which is associated with variable (tag) "v2".
3
v1
v2
S
Move the cursor to a coil "v2" and press the space bar twice to change the coil type to "set" coil.
Move the cursor to the coil "v1" and press the "delete" key to delete coil "v1".
4
v2
S

5.3.3.7. Inserting and Deleting a Function Block

To demonstrate how to insert or delete a function block and other related issues, go through the following steps.

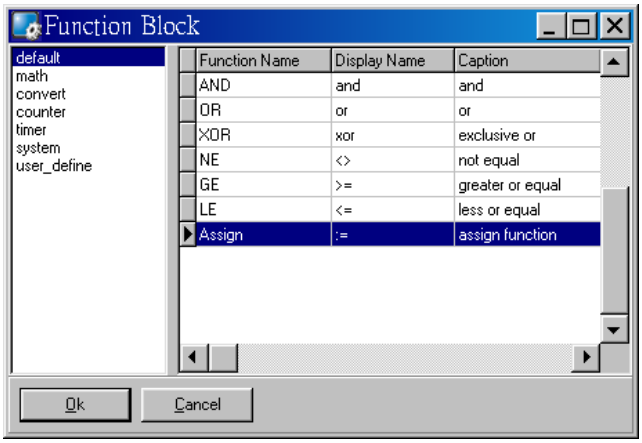
1. Set the function type to a function block
   i. Insert a new rung
      Press **F6** to insert a new rung with a function block and a coil output.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



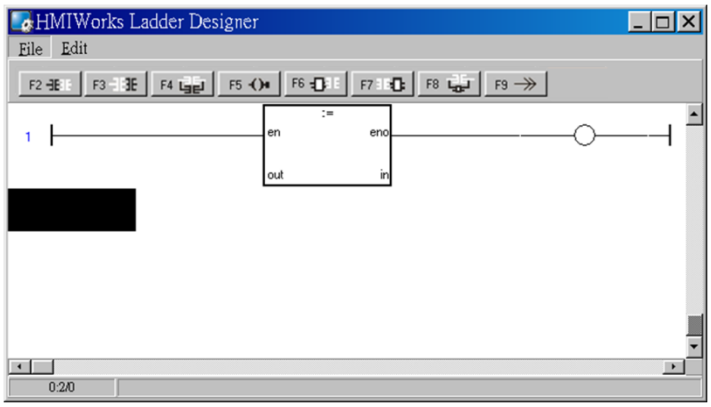
ii. Choose function type

In the new rung, double-click on the function block to open the "**Function Block**" window.



Double-click on the "Function Name" field in the list to set the type of the function.

For example, we double-click on the Function "Assign" in the default group and set to the function block.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

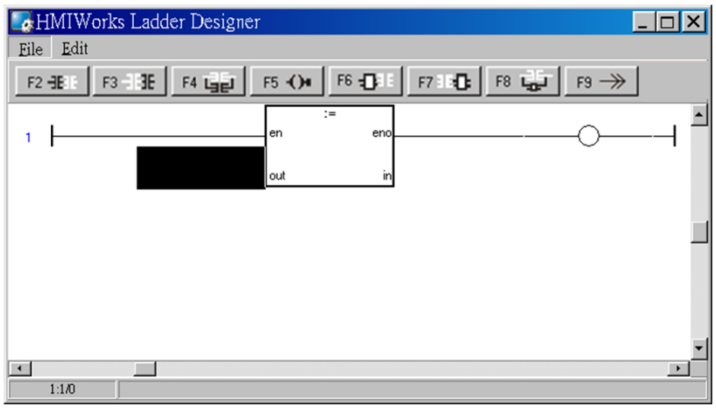
iii. Assign the variables to the function

Now, we should assign the variable to the function “Assign”. As you can see, there are four variables, en, eno, out, in.

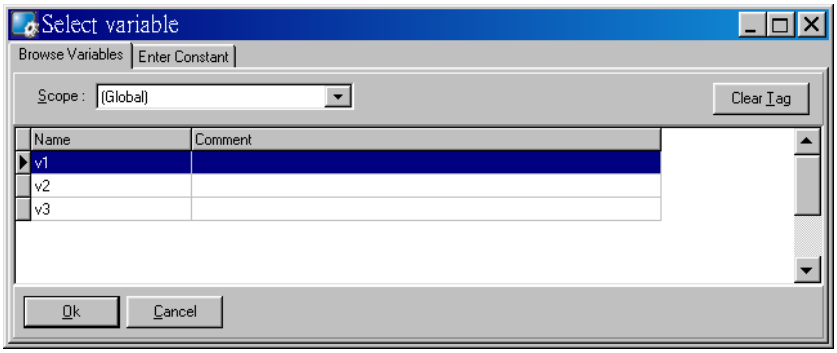
- Both “en” and “eno” cannot associate variables by users.
- We can associate “out” and “in” with the variables we define by “**New Virtual Tags**”.

For example, we associate “v1” to “out” and “v2” to “in”. v1, v2 and v3 are the variables defined in from the “**Edit Variable**” dialog box. Refer to the “**New Virtual Tags**” section.

To associate “v1” to “out”, move the cursor just beside “out” but not in the function block.

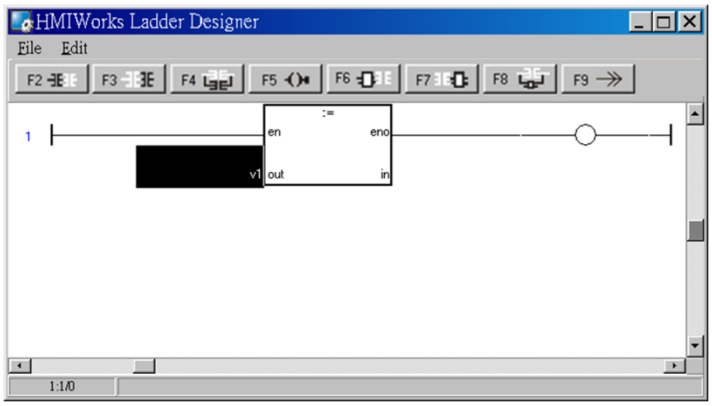


Double-click on **just beside “out”** to open “**Select variable**” window.

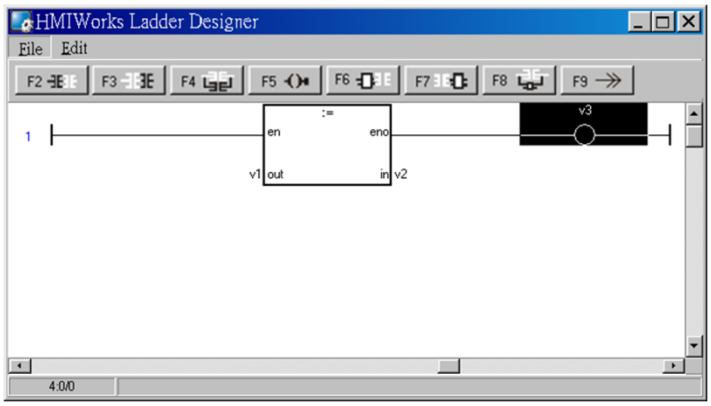


Double-click on the variable in the list to assign the variable to “out”. For example, we double-click on the variable “v1” and set to “out” of “Assign” function.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



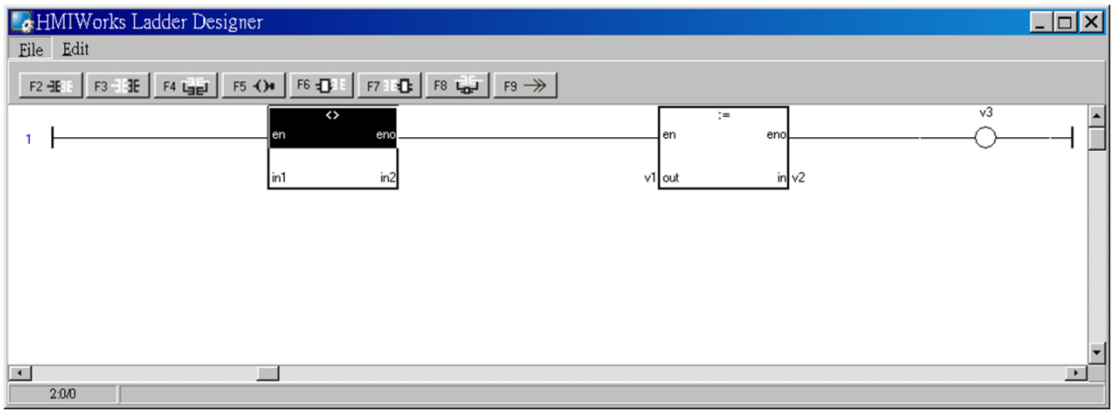
Set “v2” to “in” of “Assign” function in the same way.
Finally, set “v3” to the coil output.



This function assigns “v2” to “v1” if en is set to high.
The coil output “v3” is purely defined by eno, where eno = en.

2. Insert a new function block in the left of the cursor (**F6**)

Move the cursor to the “Assign” function block and then press **F6**.
And to make things clear, set the newly-inserted function block as “NE” (not equal).

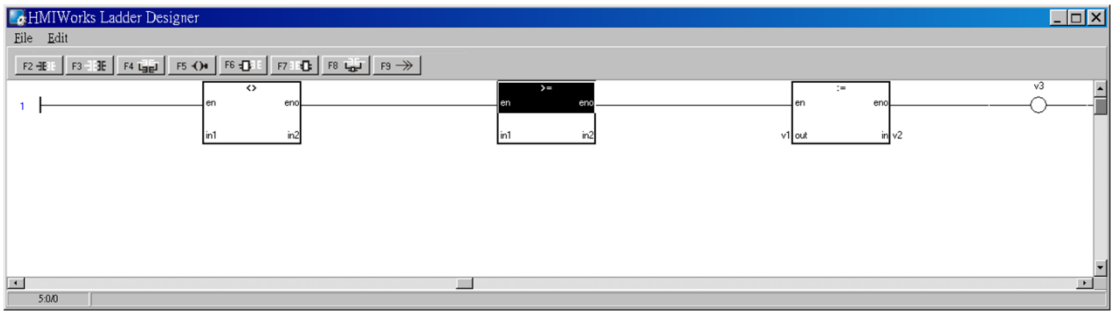


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

3. Insert a new function block in the right of the cursor (**F7**)

Move the cursor to the “NE” function block and then press **F7**.

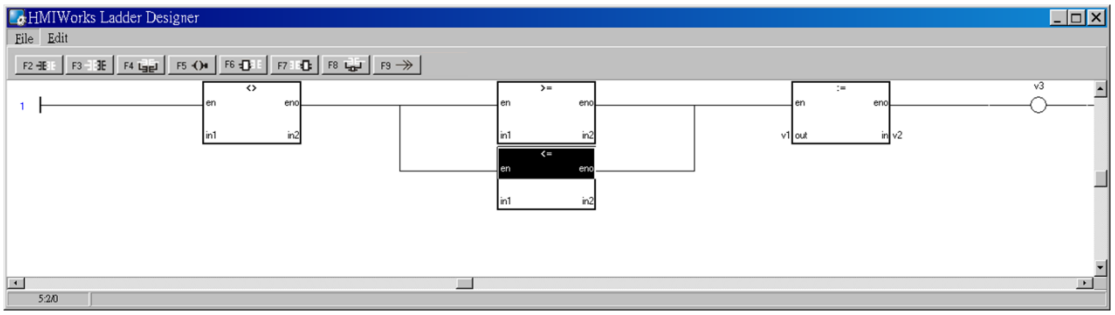
Set the newly-inserted function block as “GE” (greater than or equal).



4. Insert a new function block which is parallel to the cursor (**F8**)

Move the cursor to the “GE” function block and then press **F8**.

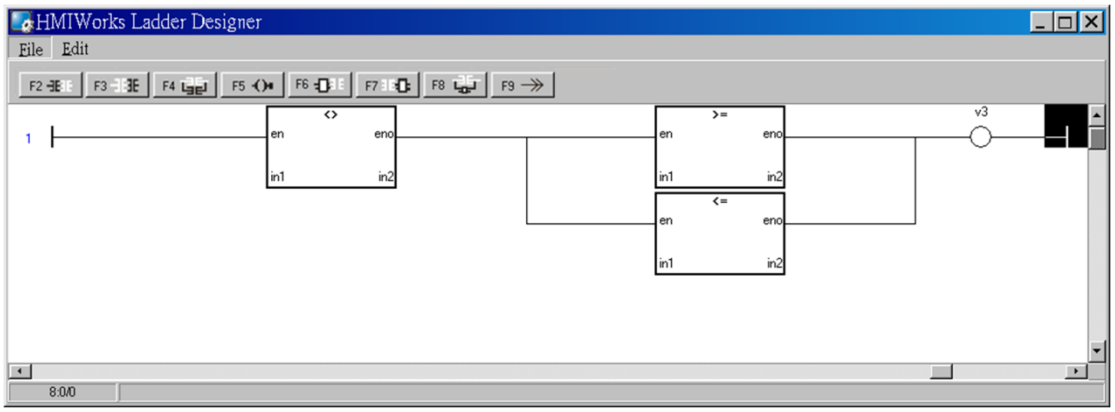
Set the newly-inserted function block as “LE” (less than or equal).



5. Delete a function block in the rung

Move the cursor to the function block you want to delete. Then press “**delete**” on your keyboard.

For example, we move the cursor to the “Assign” function block and then press the “**delete**” key.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

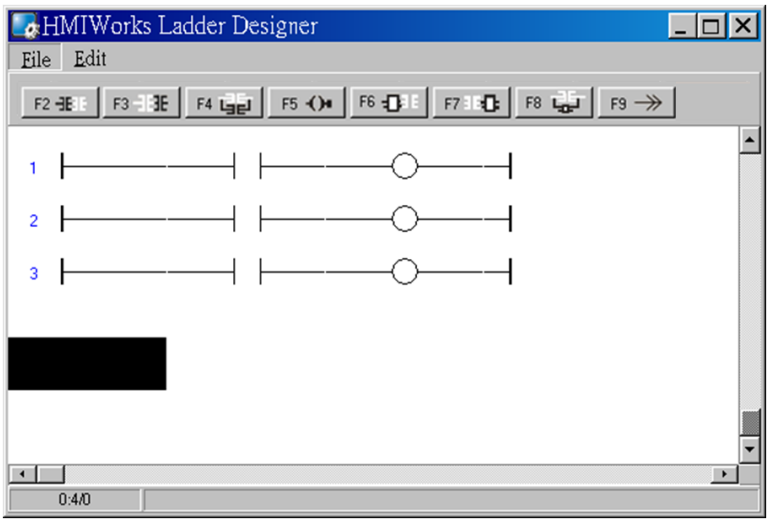
6. Delete the rung.

Move the cursor to the starting point of the rung and then press “**Delete**” on your keyboard.

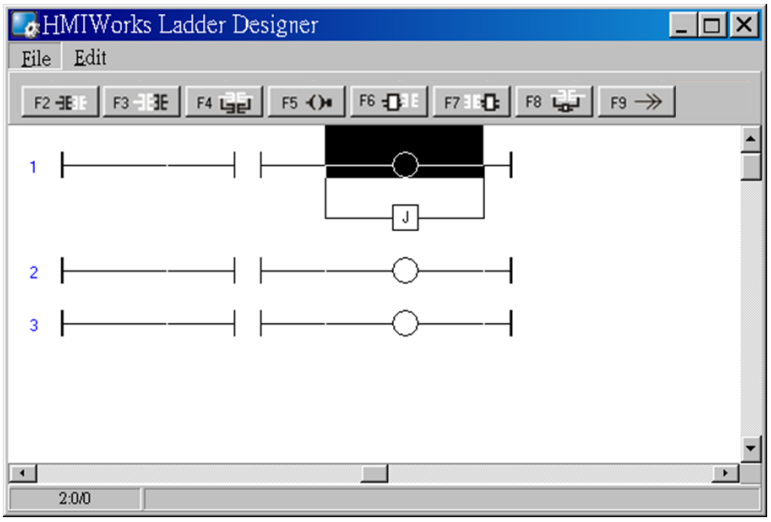
5.3.3.8. Jump to a Label

To demonstrate how to jump to a label, first we create three rungs and then explain how to skip the second rung and jump to the third.

1. Press **F2** three times to create three rungs for example.



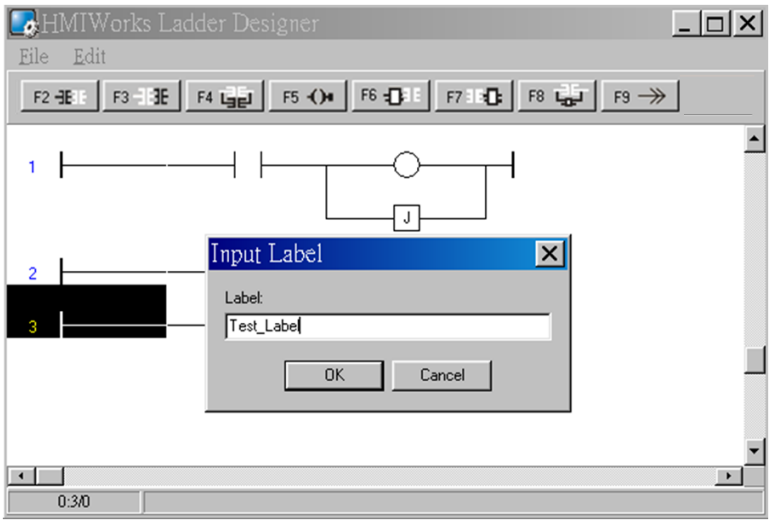
2. Move the cursor to the coil output of the first rung and then press **F9** to add a Jump.



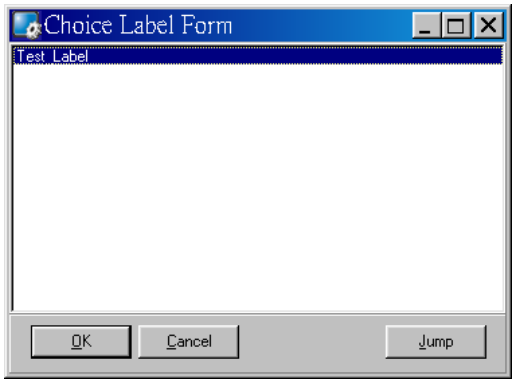
3. Double click on the starting point of the third rung to add a label

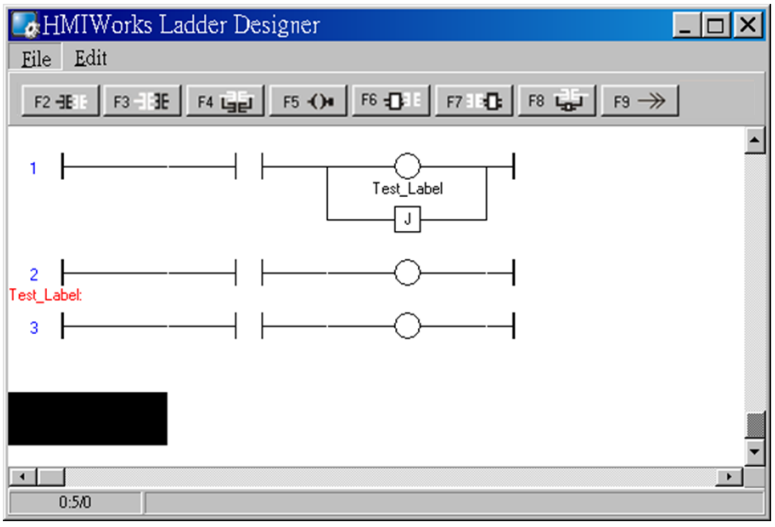
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

“Test_Label” to it.



4. Double click on the Jump of the first rung to associate with the label of the third rung.





5. When running the ladder logic, set the coil output of the first rung to high, skip the second rung and jump to the third rung if the contact input of the

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

first rung is closed.

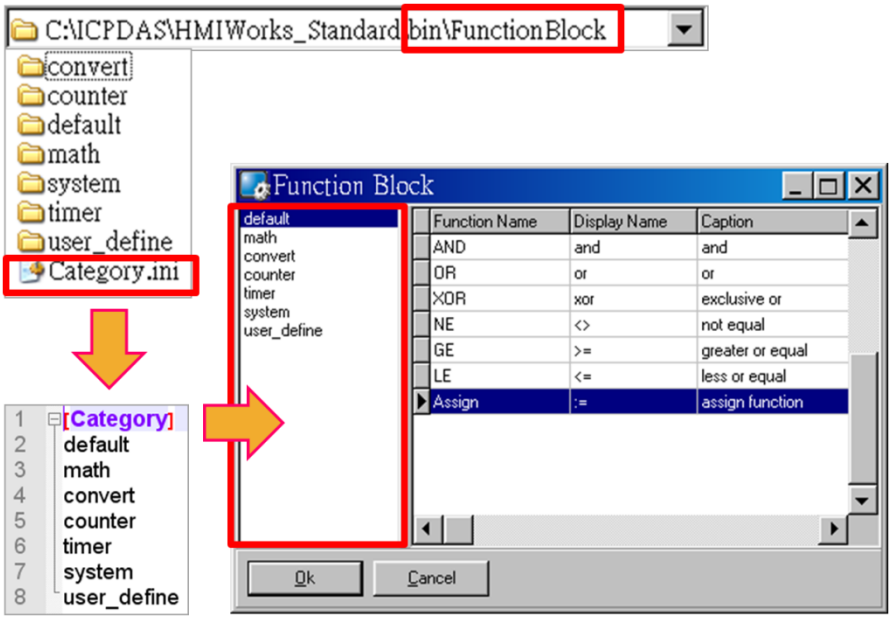
5.3.4. User-Defined Function Block

Why should we use function block?
There may be cases that using only ladders is too complex. At that time, using a function block may be a good choice.

To know how to add a user-defined function block, we first explain how HMIWorks uses these function blocks. Take “Assign” function block in the “default” group for example.

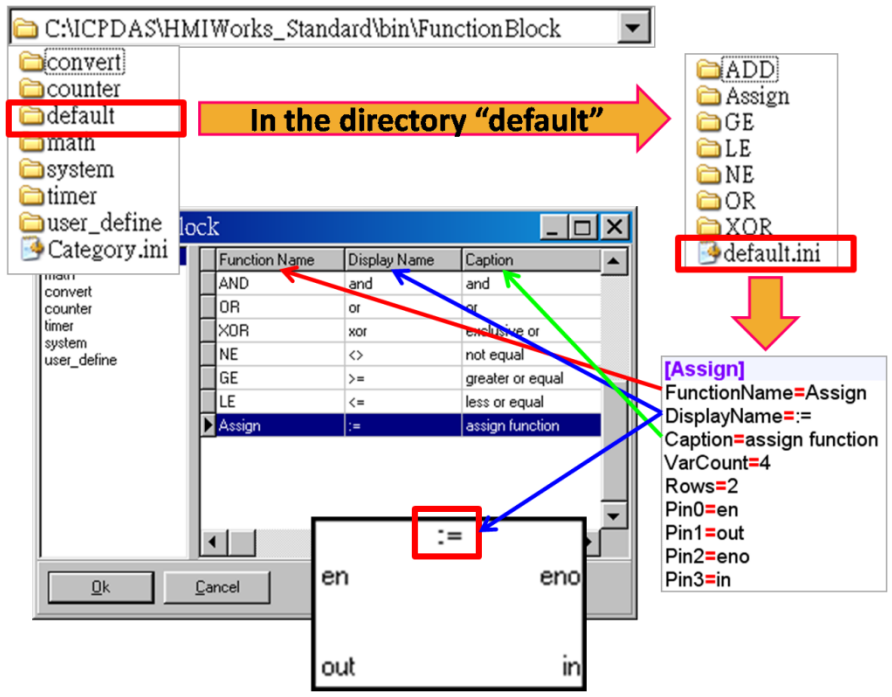
How HMIWorks Uses Function Blocks

1. Go to the installation path of the HMIWorks software. In the sub-directory, “bin\FunctionBlock”, of that installation path, open the file “Category.ini” to load the groups.

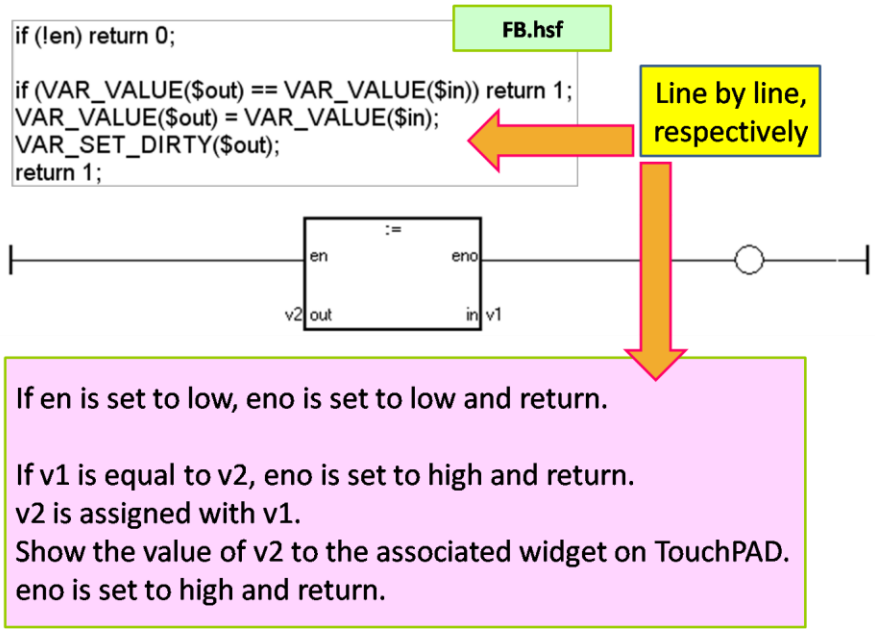


2. If we choose the “default” group, then HMIWorks opens the matching-name sub-directory and then loads from the matching-name “.ini” file in that sub-directory. That is, the “default.ini” in the sub-directory “default”.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



3. Double click on the "Assign" to use it in the **Ladder Designer**. The **Ladder Designer** uses the logics defined in the file "FB.hsf" in the sub-directory "Assign". FB.hsf is based on the C language. The following figure explains what FB.hsf of the "Assign" function does.


FB.hsf
if (!en) return 0;
if (VAR_VALUE($out) == VAR_VALUE($in)) return 1;
VAR_VALUE($out) = VAR_VALUE($in);
VAR_SET_DIRTY($out);
return 1;
Line by line, respectively
:=
en
eno
v2
out
in
v1
If en is set to low, eno is set to low and return.
If v1 is equal to v2, eno is set to high and return.
v2 is assigned with v1.
Show the value of v2 to the associated widget on TouchPAD.
eno is set to high and return.

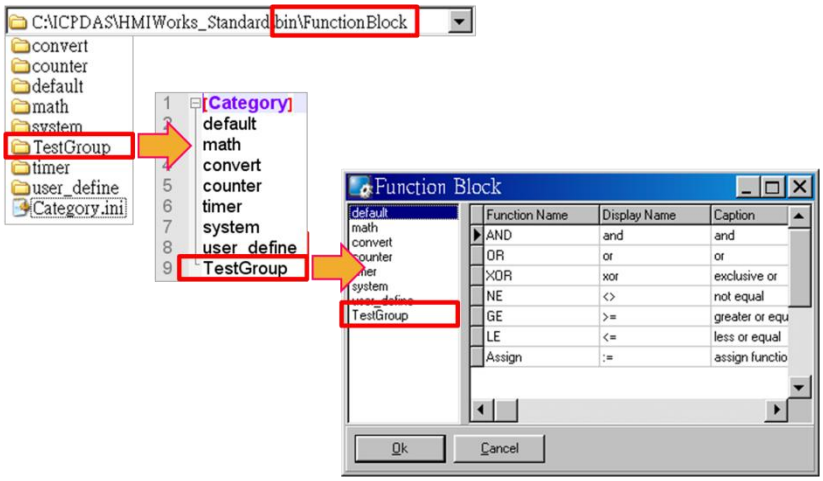
Adding a User-Defined Function Block

Now, we introduce how to add a user-defined function block.

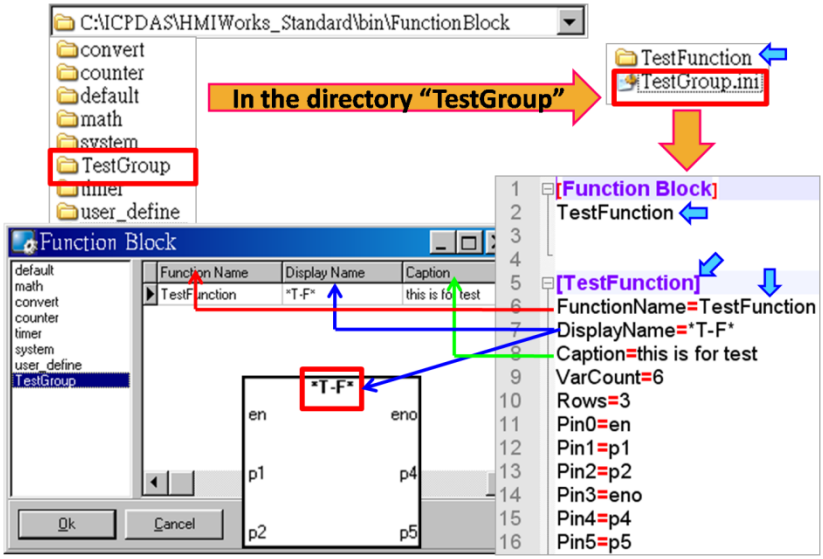
1. Create a new group if necessary.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Go to the installation path of HMIWorks. In the sub-directory of “bin\FunctionBlock”, create a new directory “TestGroup” for example and open the file “Category.ini” to add a new item to represent the new group.
Note: The name of the new item in the Category.ini **must** be exactly the same as the name of the newly-created directory.

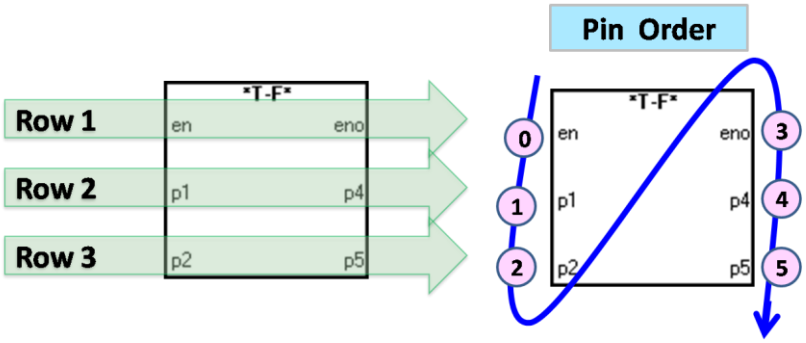


2. Go to the directory “TestGroup”, create a .ini file of the exactly same name as that of the group, that is, “TestGroup”. Create a sub-directory of the “TestGroup” directory and we may call the sub-directory “TestFunction”. Finally, define a new function, “TestFunction” in the file “TestGroup.ini”.



Note: VarCount = pin counts.
Below shows what does the Row mean and the order of the pins.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



3. In the directory “TestFunction”, create a new file FB.hsf to implement the user-defined function.

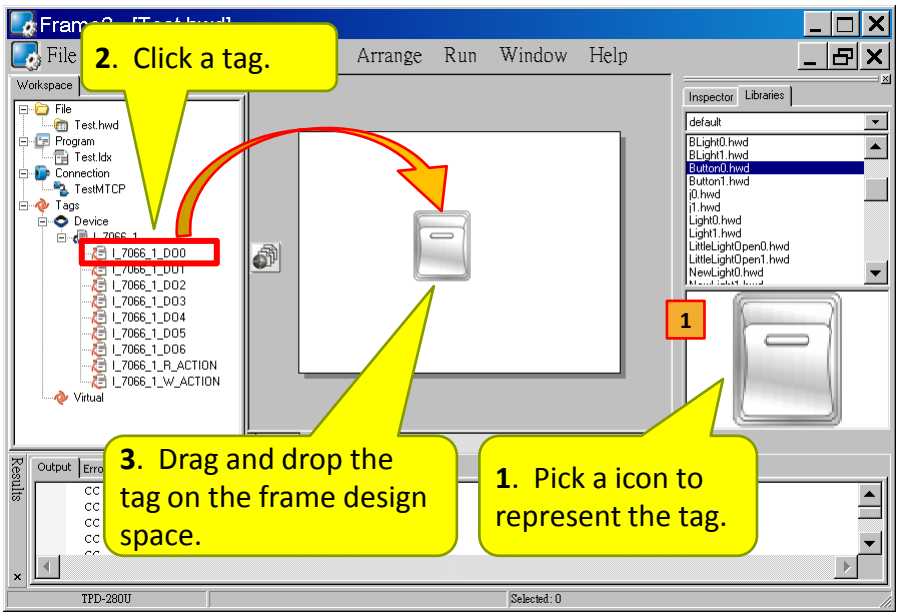
5.3.5. Associate Tags with Tools

In order to use **Ladder Designer** to build HMI of TouchPAD, we should associate tags with tools.

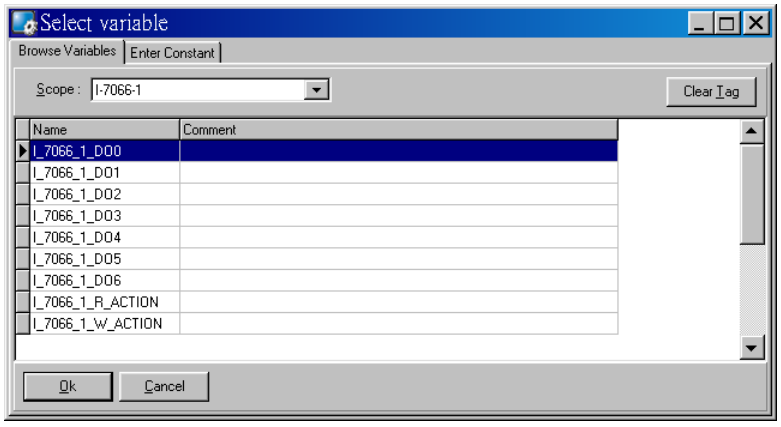
There are three methods to associate tools with tags. Every change of the tag in the **Ladder Designer** is updated to the tool in the run time after association.

1. The first method: simply drag and drop the tags in the **Workspace** to the frame design area. A CheckBox component is created with the tag associated.
   Note: this feature is only supported for the CheckBox components.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

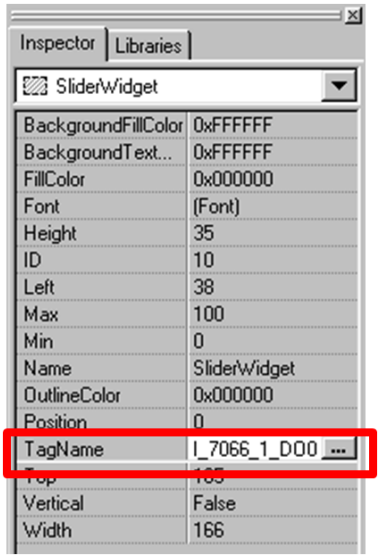


2. The second method: double click the widget on the frame design area to open the "**Select variable**" window. Take a Slider for example.



Double click on the tag Name you want to associate with the widget. Then you can see the tag is associated with the widget (that is, the Slider for example) by setting the property TagName to the name of the tag.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



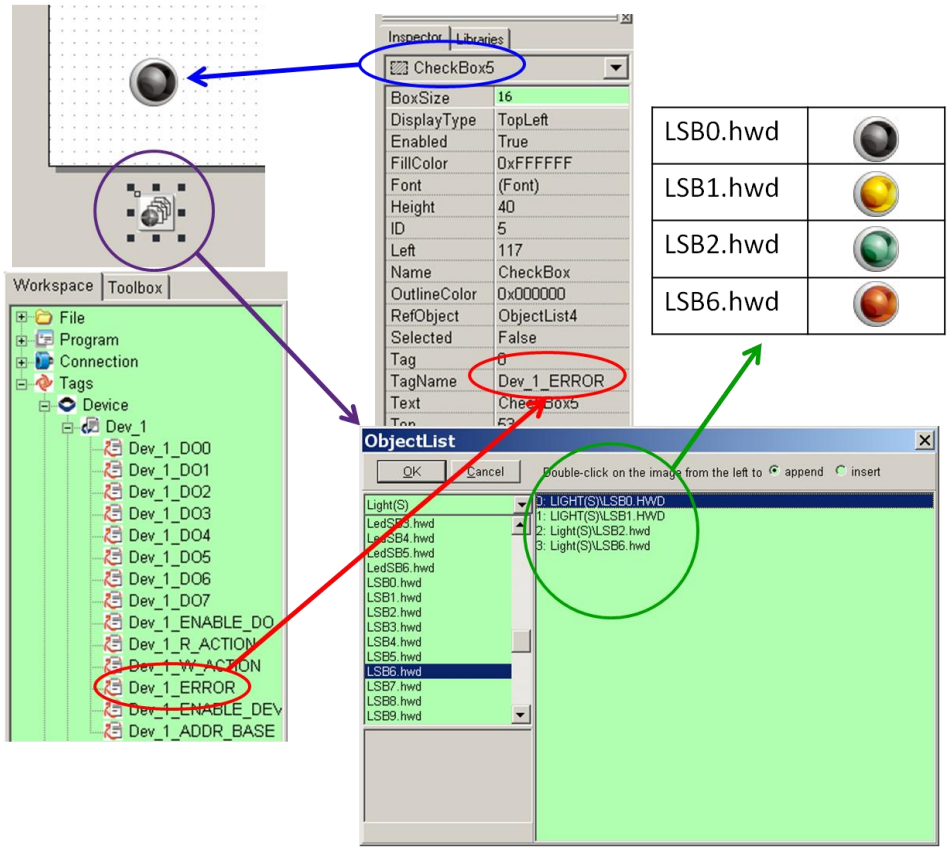
3. The third method is click the "**…**" button from the TagName field in the Inspector to open the "**Select variable**" window. Similar steps as above.

Special Note:
Refer to section "Using an ObjectList". Set the RefObject property of a CheckBox component to an ObjectList component which contains images and then associate a tag to the CheckBox component. Then every time the tag changes its value, the CheckBox component toggles the images. This feature is especially useful when building switches.

Connecting Blinking Cycle
Used for communications of Modbus TCP master polling (remote slave devices) ONLY, the Connecting Blinking Cycle defines the blinking period of "ERROR" tag used in devices which can be found in the Workspace.

As shown below, this figure demonstrates the usage of Connecting Blinking Cycle. A CheckBox is used to be a signal of communication status of a remote Modbus TCP slave device and is assigned an ObjectList of four images. (Of course, TouchPAD must be a Modbus TCP master device.) To compatible with the old versions of HMIWorks, the first and the second images must represent "communication normal" (connected) and "communication error" (disconnected). The third and the fourth images toggle when TouchPAD is in its connecting status. The period of connecting blinking can be found in the "Project Configuration" form in the HMI menu.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



5.3.6. User-Defined I/O Modules

To know how to add a user-defined I/O module, we first explain how HMIWorks uses these I/O modules.

There are several kinds of I/O modules.

- DCON I/O modules: I-7000 series I/O modules by ICP DAS.

http://www.icpdas.com/products/Remote_IO/i-7000/i-7000_introduction.htm

- Modbus TCP I/O modules: ICP DAS provides ET/PET-7000 series.

http://www.icpdas.com/products/Remote_IO/et-7000/et-7000_introduction.htm

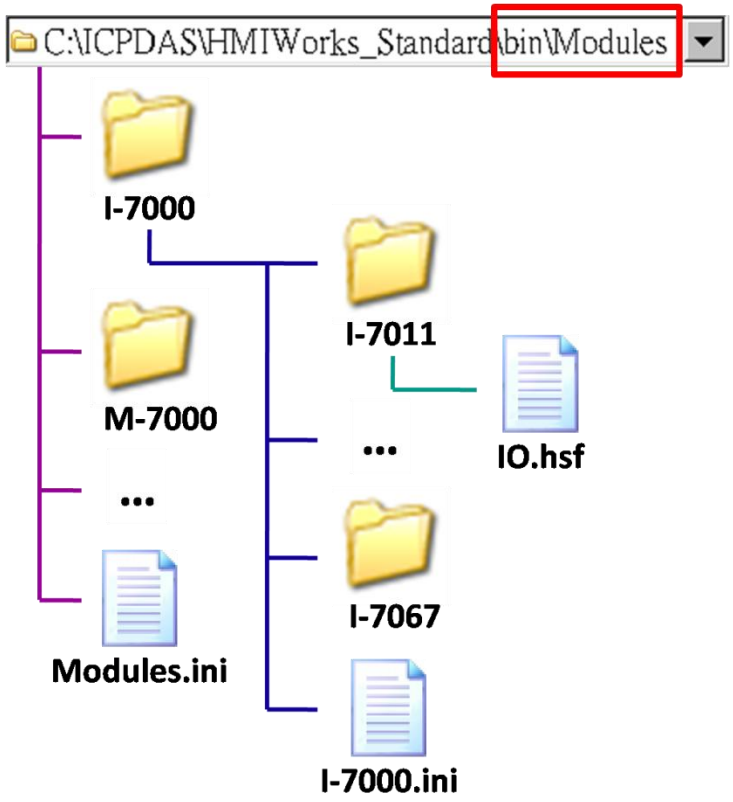
- Modbus RTU I/O modules: M-7000 series I/O modules by ICP DAS

http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_introduction.htm

Where HMIWorks Put I/O Module Information

HMIWorks puts I/O module information in the following locations.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

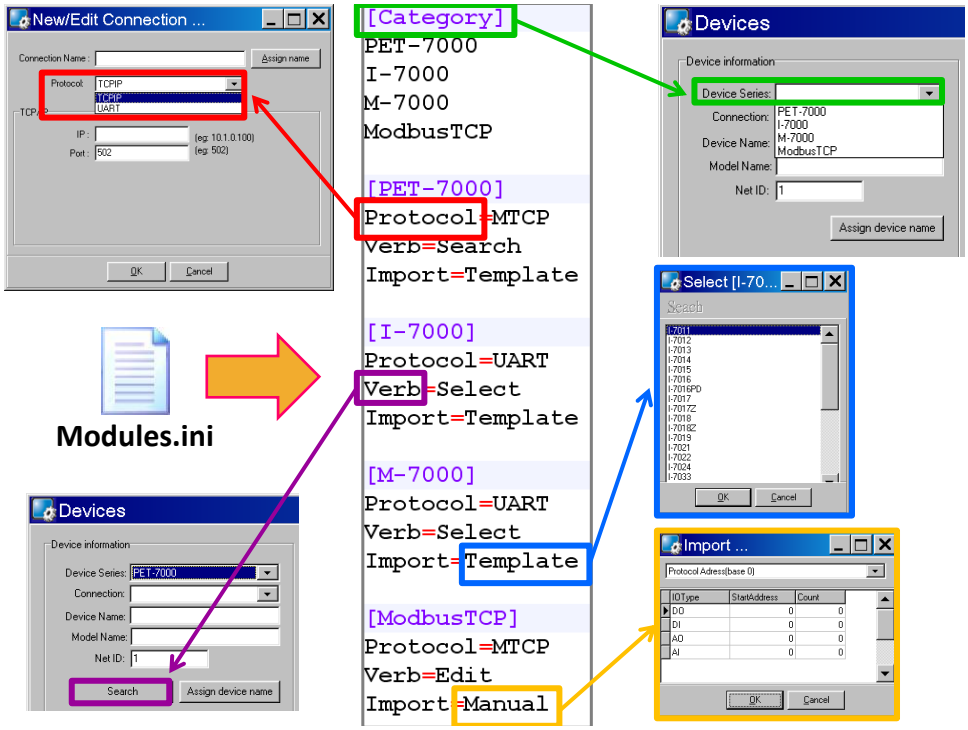


Some explanations for above figure:

- “C:\ICPDAS\HMIWorks_Standard\” is the installation path. (Users may have different installation paths.)
- “Modules.ini” is the I/O series configuration file.
- “I-7000.ini” is the configuration file for the I-7000 series I/O modules. “M-7000.ini” and “PET-7000.ini” are configuration file for the M-7000 and the PET-7000 series I/O modules respectively.
- Each I/O module has a matching name directory and in that directory there is only one file, IO.hsf. IO.hsf is the file of the C language to define the behaviors of the I/O module.
- I/O module directories in the same series are grouped together in the I/O series directory. For example, I-7011, …, I-7067 are directories represent I/O modules and they are all put to the series directory “I-7000”.

What Module.ini describes?

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



In details, we have the following table:

Item		Description
Category		This section keeps the list of the device series which HMIWorks supports. When registering device (**F3**), the "**Devices**" window gets the information of device series from this "Category" section.
Protocol	MTCP	"Protocol=MTCP" in the Module.ini is corresponding to "Protocol=TCPIP" in the "**New/Edit Connection**" window in the **Workspace**.
	UART	"Protocol=UART" in the Module.ini is corresponding to "Protocol=UART" in the "**New/Edit Connection**" window in the **Workspace**.
Verb	Search	HMIWorks scans through the network to find out I/O modules. Until now, PET-7000 is the only series which support this "Search" function.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

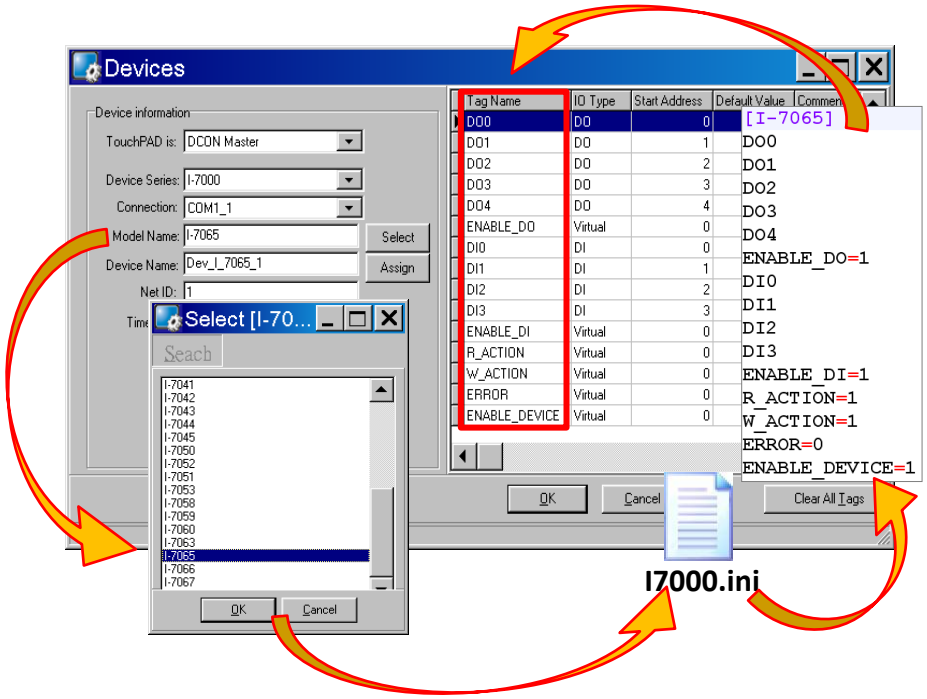
	Select	HMIWorks pops up a list of I/O modules to let users select one. The list of I/O modules is loaded from the file whose name is [Device_Series_Name].ini
	Edit	HMIWorks opens the "**Import**" window to let users decide the I/O points for the I/O module.
Import	Template	HMIWorks imports the tags of the I/O module from the I/O module configuration file. For example, HMIWorks imports tags of I-7011 from the template in the file of I-7000.ini.
	Manual	HMIWorks imports the tags of the I/O module by the manually-decided I/O points.

Generating Tags by "Register Devices (F3)"

Press **F3** on your keyboard to open the "**Devices**" window to register I/O devices.

The I/O modules configuration file has templates for all the I/O modules in the I/O series. For example, I-7000.ini is the configuration file for the I-7000 series I/O modules.
Take I-7065 in the I-7000 series for example as shown in the following figure.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



As above, ERROR is the tag for the communication status.

Defining I/O Behaviors in IO.hsf

- Take I-7065 for example (I-7000 series I/O module)
  Open the IO.hsf in the directory "[HMIWorks install path]\bin\Modules\I-7000\I-7065\".
  The codes in IO.hsf are of C language as below:

```
BEGIN_FUNCTION_BLOCK(); //this line is necessary

DWORD v_do = 0;
DWORD v_di = 0;
int     gWriteCount = 0;

uart_SetTimeout($DEVICE, $TIMEOUT);

//$W_ACTION: a tag used in Ladder to enable/disable writing actions
//$ENABLE_DO: a tag used in Ladder to enable/disable the part of DOs
if ( VAR_VALUE($ENABLE_DO) && VAR_VALUE($W_ACTION))
{
    int iWrite = 0; //To decide if there's a need to write any DO channel
```

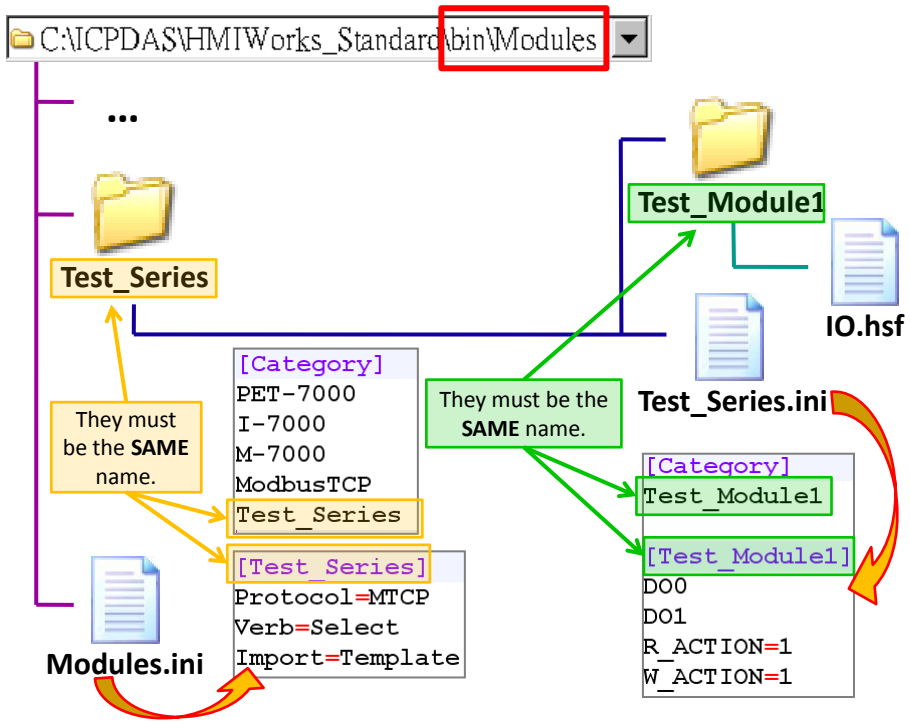

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

```
    v_do     = 0;

    // Update the status for each channel if it has been changed.
    iWrite += VAR_GET_WRITE_U32(&v_do, $DO0, 0);
    iWrite += VAR_GET_WRITE_U32(&v_do, $DO1, 1);
    iWrite += VAR_GET_WRITE_U32(&v_do, $DO2, 2);
    iWrite += VAR_GET_WRITE_U32(&v_do, $DO3, 3);
    iWrite += VAR_GET_WRITE_U32(&v_do, $DO4, 4);

    if ( iWrite )    // Write only when need
    {
        gWriteCount++;
        if ( ! dcon_WriteDO($DEVICE, $NETID, 5, v_do & 0xFF) )
        // dcon_WriteDO: the DO writing API function of I-7000 I/O series.
        // I-7000 I/O series uses the DCON protocol.
            return HMI_ERROR;
    }
}

if ( gWriteCount ) return HMI_OK;
// Skip reading to reduce the device loading

if ( (VAR_VALUE($ENABLE_DO) || VAR_VALUE($ENABLE_DI)) &&
VAR_VALUE($R_ACTION)) {
//$R_ACTION: a tag used in Ladder to enable/disable reading actions
//$ENABLE_DO: a tag used in Ladder to enable/disable the part of DOs
//$ENABLE_DI: a tag used in Ladder to enable/disable the part of DIs
if (dcon_ReadDIO($DEVICE, $NETID, 4, 5, &v_di, &v_do))
// dcon_ReadDIO: the DI/DO reading API function of I-7000 I/O series.
// I-7000 I/O series uses the DCON protocol.
{
    VAR_SET($DI0, v_di & (1<<0));
    // VAR_SET: used to set the value of this channel to its tag
    VAR_SET($DI1, v_di & (1<<1));
    VAR_SET($DI2, v_di & (1<<2));
    VAR_SET($DI3, v_di & (1<<3));

    VAR_SET($DO0, v_do & (1<<0));
```


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

```
        VAR_SET($DO1, v_do & (1<<1));
        VAR_SET($DO2, v_do & (1<<2));
        VAR_SET($DO3, v_do & (1<<3));
        VAR_SET($DO4, v_do & (1<<4));
    } else
        return HMI_ERROR;
    }

    END_FUNCTION_BLOCK(); //this line is necessary
```

Creating a User-Defined I/O Module



1. In the directory, “[HMIWorks install path]\bin\Modules\”, create a new I/O series directory whose name is “Test_Series” and be sure to update Modules.ini to notify HMIWorks that there is a new I/O series called “Test_Series”. As the figure shows, the series directory name and the name in the Modules.ini must be **the same**.
2. In the I/O series directory, “Test_Series”, we create a new I/O module directory whose name is “Test_Module1” and be sure to create a I/O modules configuration file, Test_Series.ini, to depict the template of the newly-created I/O module, Test_Module1. As the figure shows, the

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

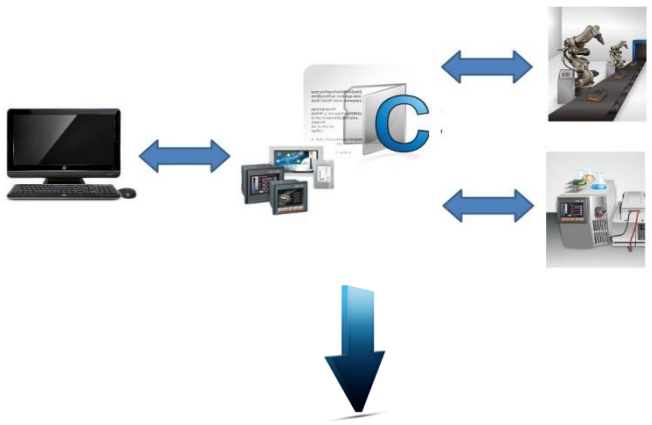
module directory name and the name in the Test_Series.ini must be **the same**.

3. Implement the IO.hsf which is created in I/O module directory, Test_Module1, to describe the behaviors of the I/O module, Test_Module1. Refer to the IO.hsf
   I. of PET-7000 series if using the Modbus TCP protocol.
   II. of M-7000 series if using the Modbus RTU protocol.
   III. of I-7000 series if using the DCON protocol.
   All are similar to the example of the I-7065 above.

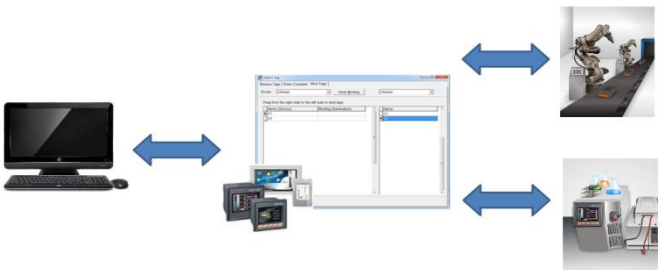
5.3.1. Data exchange

Uniform standards data format by the TouchPAD and served as the role of protocol conversion to the exchange of information between the different agreements and resolve master and slave exchange of information between the problem of data transfer between the device to automatically "Agreement", "Handle "and" Respond "and let live applications more flexible.

Use C code to convert is so complexity

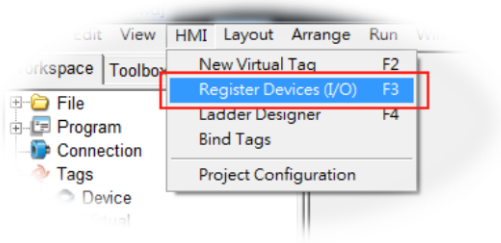


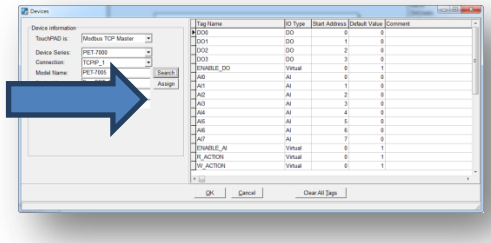
And use data exchange function after, it’s so easy.



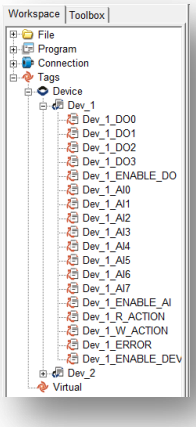
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

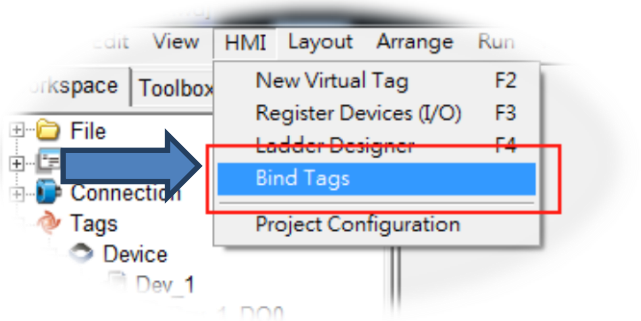
Step1. Add device in HMIWorks.





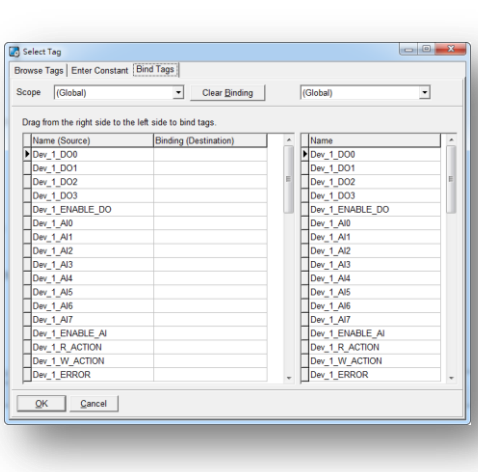
Step2. Add more devices after,you can see the workspace field and then add tag. And then go to HMI-> Bind Tags

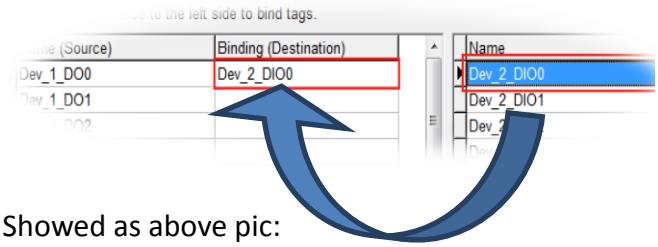




Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Step3. Drag the tag that make the relatedness, and the instructions for use as below.





Showed as above pic:

1. Drag the tag from right to left.

2. When the Dev_2_DIO0 drag to Dev_1_DO0 :

If Dev_1_DO0 changed, then the Dev_2_DO0 value will upade to Dev_1_DIO0 value:
Dev_2_DO0= Dev_1_DIO0

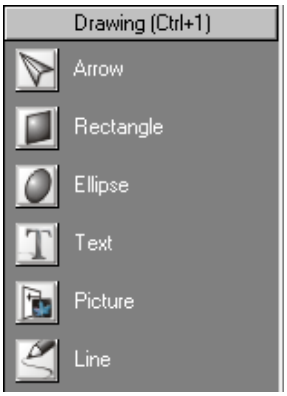
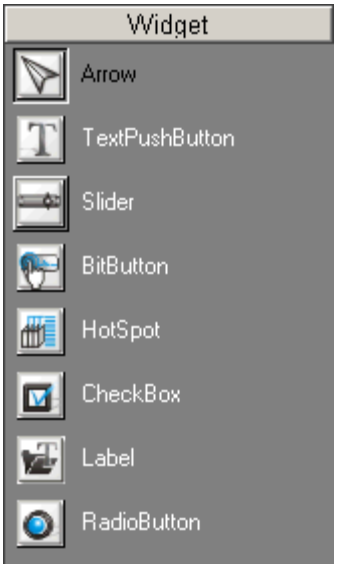
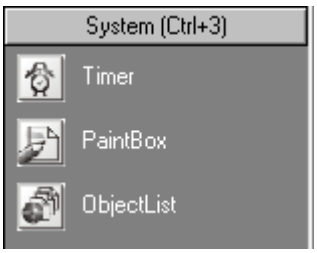
3. For example, when B drag to A, C drag toB, if A changed, then B=A,C=A

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4. Frames and Components

This section introduces properties and usages of frames and components from the **Toolbox**.

In the **Toolbox**, there are three kinds of components, the Drawing, the Widget and the System components.

<table>
<tr>
<td> 
</td>
<td>Drawing:
1. Rectangle: draw a rectangle.
2. Ellipse: draw ellipse.
3. Text: put string (text) on screen.
4. Picture: load an image file on a frame.
5. Line: draw a line.</td>
</tr>
<tr>
<td> 
</td>
<td>Widget:
1. TextPushButton: create a button.
2. Slider: show or decide the percentage.
3. BitButton: create an image button.
4. HotSpot: create a hot spot that can issue an OnClick event.
5. CheckBox: provide an alternative.
6. Label: provide a string that can be modified during the run-time.
7. RadioButton: provide a “one-of-many” selection</td>
</tr>
<tr>
<td> 
</td>
<td>System:
1. Timer: periodically execute codes.
2. PaintBox: draw shapes in the run time.
3. ObjectList: maintain a list of library objects which can be used through property “RefObject” of TextPushButton and CheckBox.</td>
</tr>
</table>

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Important Notice

1. Make sure that widget component should not overlap or unexpected behavior may happen when clicking.
2. The minimum gap between two components is 12 pixels. If the gap is smaller than 12 pixels, pressing one component may trigger the other's event handler due to calibration accuracy.

5.4.1. Commons of Components and Frames

This section describes the common characteristics of frames and components from the **Toolbox**.

Putting a component on the frame

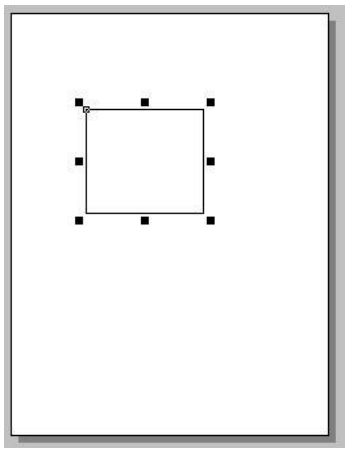
Two ways to put a component on the frame:

1. drag a suitable sized rectangle
2. simply click on the frame to decide the location after selecting a component

To drag a suitable sized rectangle, take a Rectangle for example to describe how to put a component (such as a Rectangle, a Slider, etc.) on the frame.

I. Click the Rectangle icon from the Drawing panel of the **Toolbox** tab.
II. Move your mouse to the frame design area and click and drag a suitable sized rectangle.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



P.S.

To draw a square

What to do if I want to draw a square?
Step II with the "**Ctrl**" key pressed at the same time when drawing a Rectangle.

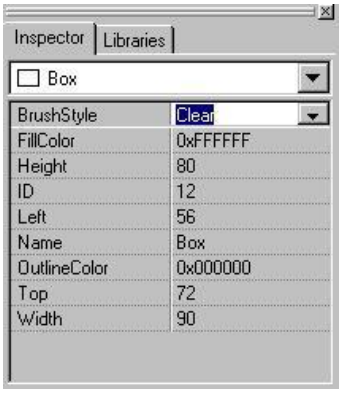
To draw a circle

What to do if I want to draw a circle?
Step II with the "**Ctrl**" key pressed at the same time when drawing an Ellipse.

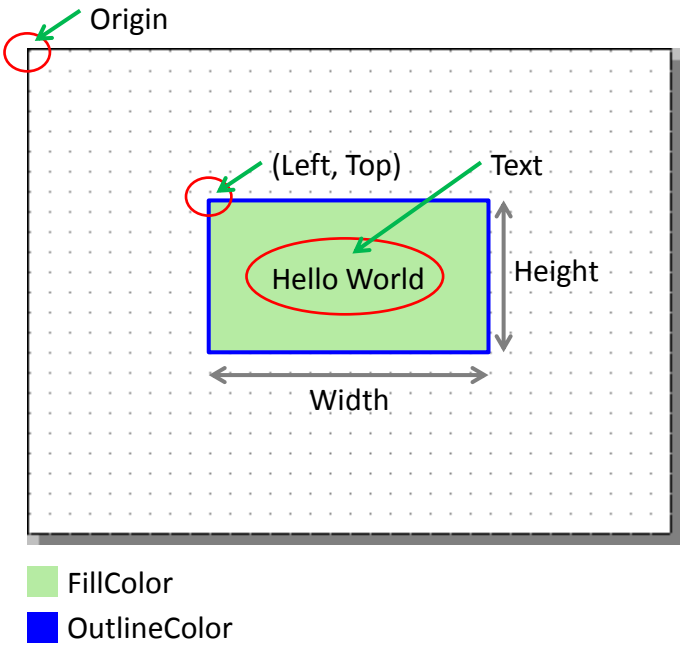
Common Properties

Where can we access properties of a component?

Click on the component (or the frame) and then the properties of the component can be accessed in the **Inspector**.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Property	Description
FillColor	The color used to fill the rectangle which encloses the component. The color is represented by a three byte value in the hexadecimal form. From the highest byte to the lowest, it is the blue, the green, and the red byte in sequence.
OutlineColor	The outline color of the rectangle which encloses the component
Height	The length of the vertical side of the rectangle which encloses the component
Width	The length of the horizontal side of the rectangle which encloses the component
Left	The x-coordinate of the left-top vertex of the rectangle which encloses the component
Top	The y-coordinate of the left-top vertex of the rectangle which encloses the component
Name	The name of the component
ID	The serial numbers of the components in the Toolbox and of the frames. These serial numbers are used to identify them.
Font	The font of the Text property

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

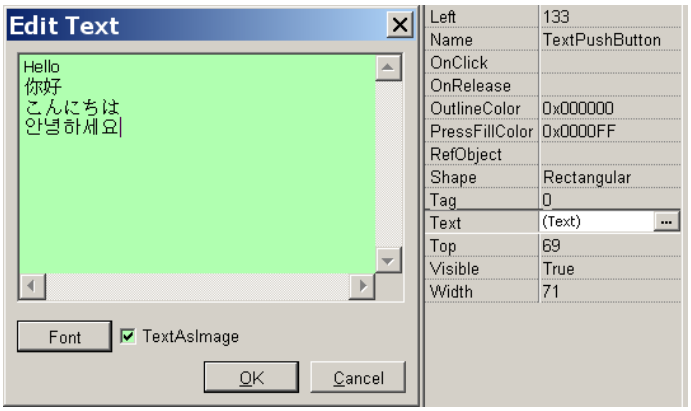
Text	The strings of the component to be displayed
GoToFrame	Go to the specified frame. That is, pressing on the owner of this property switches to the frame which is specified in this property. Note: the **priority** of the property “GoToFrame” is higher than that of “OnClick”.
RefObject	The reference to the specified object list. An ObjectList is a component that can be selected in the Toolbox to maintain a list of the images of the library. Refer to “ObjectList” section for more information.
Tag	The variable used for programming purpose. For example, it can be assigned a unique number for each TextPushButton component in order to identify them. Refer to the <<API Reference>> for functions to get/set the Tag property. Note: This Tag property has nothing to do with the “Tag” which the TagName property refers to in the Ladder mode.
TagName	Associate a variable (tag) in Ladder Designer. Note: The property is supported only in programming type “Ladder”.
Enabled	Whether the component is activated or not
Visible	Whether the component is able to be seen or not

Text into Image and Multi-language Display

There are three components (TextPushButton, CheckBox, and RadioButton) whose Text properties are not like other components and can be used to support multi-language display by transforming strings into images.

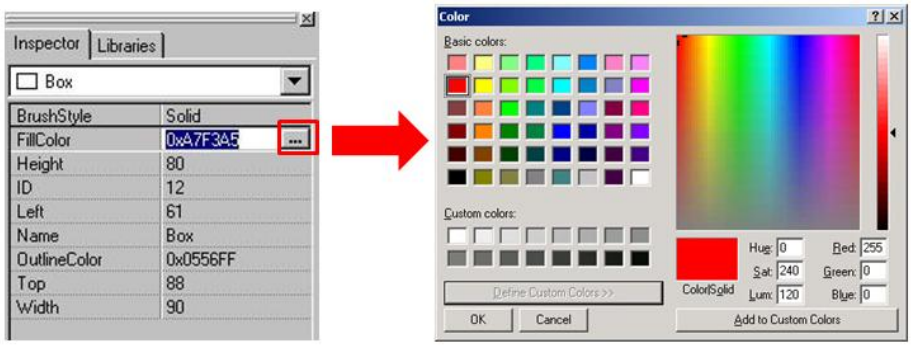
1. Check the TextAsImage option. If checked, the Text property can have multiple strings.
2. Each string of the Text property is generated into one image and each image corresponds to one state of the components. Refer to “Using the RefObject property” below for more information.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Changing the Color

To change the FillColor property of a component, click the component first to select it and then click the “FillColor” field in the **Inspector**. Then click on the “**…**” button to open the color dialog to select a color. Repeat the same procedure for the “OutlineColor” field.



Changing the Font

To change the Font property of a component, click the component first to select it and then click the “Font” field in the **Inspector**. Then click on the “…” button to open the font dialog to change the font.

There are two font dialogs when choosing fonts.

1. The same font dialog as the PC’s.
    A. If this font dialog is opened, fonts are stored as image in TouchPAD after download and therefore cost more memory space. (e.g. the same two letters, such as ‘A’ and ‘A’ cost)

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

B. Widgets that use this font dialog: Text, BitButton.

2. The custom font dialog that shows only fonts supported by TouchPAD.

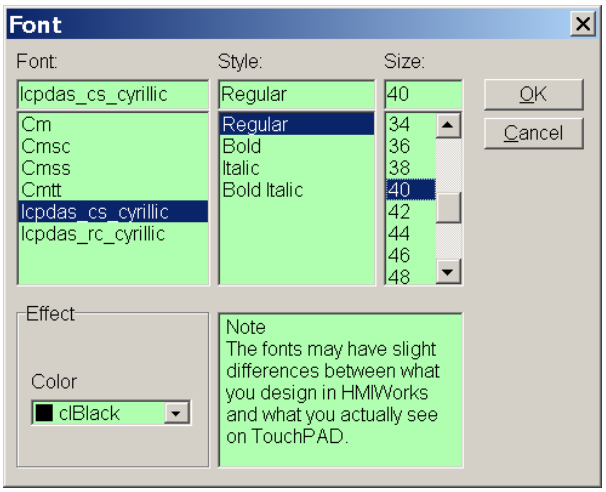
   A. The fonts does not stored as image after download. That is, the same two letters, such as ‘A’ and ‘A’, only are stored with the space one ‘A’ takes.

   B. Widgets that use this font dialog: Text, TextPushButton, Slider, Checkbox, Label, RadioButton.

   C. To support language other than English, follow the below procedure to select an appropriate font.

      a. Go to the **HMI** menu, to open the “Project Configuration” dialog. In the Font tab, select a language. (e.g. Russian)

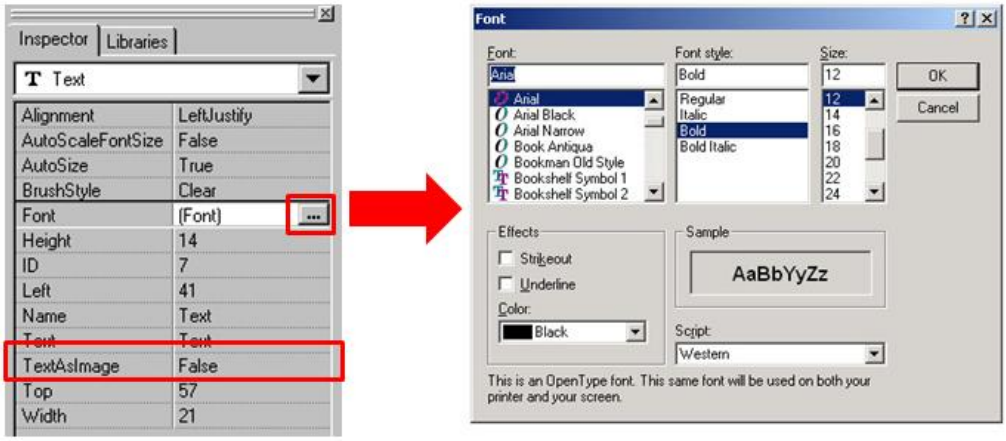
      b. Click the component in question to open the custom font dialog. Based on the language you select in the “Project Configuration” dialog, the custom font dialog displays corresponding fonts of the language. (e.g. Icpdas_cs_cyrillic, as shown below.)



Note

To use the font dialog of PC’s, the TextAsImage property of a Text component needs to be set to True.

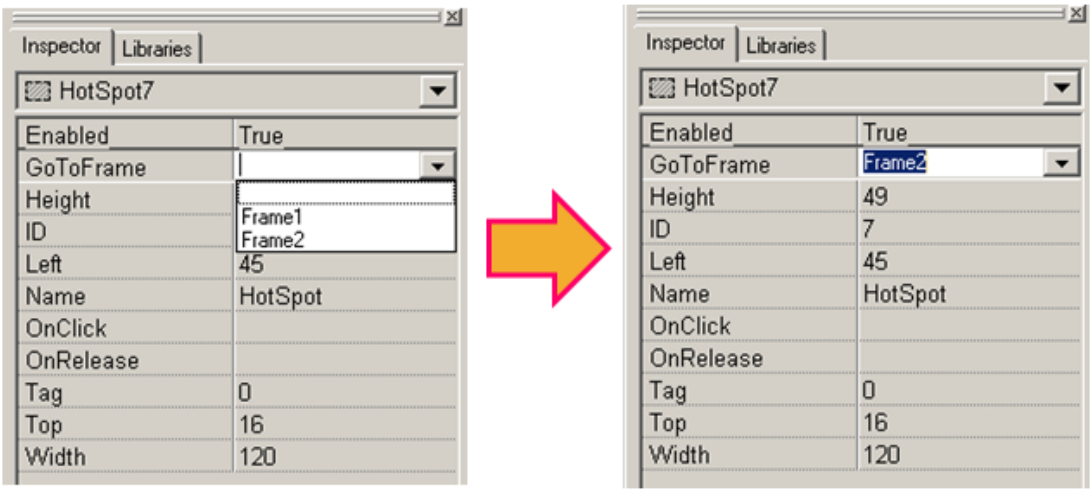
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Using GoToFrame to switch to another frame

The GoToFrame property is used as an event of go-to-specified-frame. **It has higher priority than other events, such as OnClick event.** Thus specifying an option of the GoToFrame property disables the OnClick event.

It's easy to specify a value to the GoToFrame property. Simply click the "GoToFrame" field in the **Inspector** and then choose the frame to go.



Using the RefObject property

We use the RefObject properties to replace the display of TextPushButton, Slider, CheckBox, and RadioButton with images of the assigned ObjectList. The state (or value) of a component is used as an index to determine which image in the ObjectList is displayed if the ObjectList is assigned to the RefObject property. The state can be changed by human touch, API

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

functions (e.g. CheckBoxValueSet), and tags which are specified by the TagName property.
Supposed that an ObjectList called OL is assigned to the RefObject of a component (e.g. CheckBox) and it has n images, OL[0], OL[1], …, OL[n-1].

Component /Frame	Max No. of Image	Component Behavior
Frame	1	OL[0] is the background image. Note1: Any more images in the ObjectList, OL, has no effect, they simply waste memory. Note2: Assigning the RefObject property of the default frame (the frame which has its default property equal to True) automatically assign the same ObjectList to all the frames in the project.
TextPushButton	Unlimited, theoretically	OL[0] is the background image. When the TextPushButton is in the released state, it displays the OL[0]. And when it is in the pressed state, it displays OL[1] for the first click, OL[2] for the second click after releasing the first click, and so on. While the TextPushButton reaches the last image, OL[n-1], it will start to display from the beginning again for the next click, that is, OL[1], and go on the next round.
Slider	Unlimited, theoretically	OL[0] is the background image. The Slider is divided into n-1 segments and draws the corresponding image according to the value of the Slider. See the table below for example.
CheckBox	Unlimited, theoretically	Every click on the CheckBox changes the display image, started from OL[0] to OL[n-1], one by one.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

		Once reaching the last image, OL[n-1], it restart to display from the first image for the next click, OL[0], again.
Label	1	OL[0] is the background image. Note: any more images in the ObjectList, OL, has no effect, they simply waste memory.
RadioButton	2	OL[0] is the background image. OL[1] is the selected image. Note: any more images in the ObjectList, OL, has no effect, they simply waste memory.

Slider example for the RefObject property

Example	Description
	6 Images in the ObjectList, OL. From left to right, they are OL[0], OL[1], …, OL[5].
	OL[0] is taken as a background image. The Slider is divided into 5 segment, 20% for each one, and is drawn by its value: 0% ~ 20%: OL[1] 20% ~ 40%: OL[2] 40% ~ 60%: OL[3] 60% ~ 80%: OL[4] 80% ~ 100%: OL[5] As shown in the left column.

Implementing event handlers

The event handler is supported only in the frame of C, not Ladder. By default, double clicking on the component opens the programming window of the

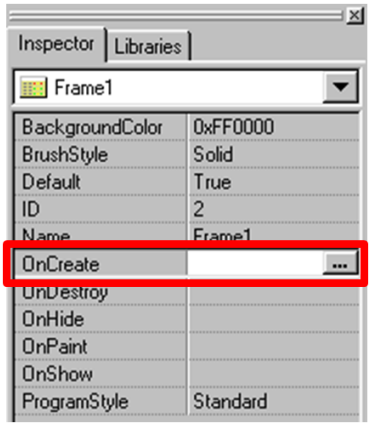
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

OnClick event handler if more than one event handlers that a component has.

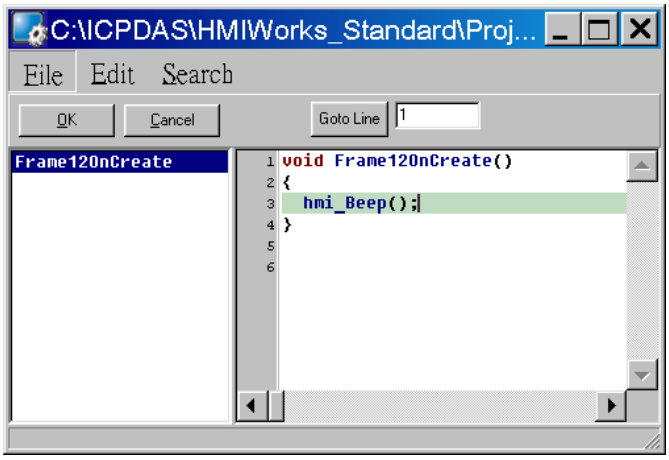
Component /Frame	Existing Event Handler
Frame	OnCreate, OnDestroy OnHide, OnShow OnPaint
TextPushButton, BitButton, HotSpot,	OnClick, OnRelease
Slider	OnSliderChange
CheckBox	OnChange
Timer	OnExecute
PaintBox	OnPaint
RadioButton	OnRadioChange

Take OnCreate event handler of a frame for example.

1. Click on the “OnCreate” field in the **Inspector**. Then click on the “**…**” button to open the programming window.



2. Here we use hmi_Beep() to sound a beep for example.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

3. Press **OK** to save the file and leave.

5.4.2. Frame

Unique Properties of a Frame

Click on the frame, and the properties of the frame are shown in the **Inspector**.

properties	description
BackgroundColor	The background color of the frame. The color is represented by a three-byte value in the hexadecimal form. From the highest byte to the lowest, it is the blue byte, the green byte, the red byte in sequence.
BrushStyle	Solid or Clear. If BrushStyle is set to “Solid”, then the setting of the “BackgroundColor” property does take effect. However this may make the **screen flash** if background color is quite different from the loaded picture. Setting BrushStyle Clear disables the “BackgroundColor” property and prevents the screen from flashing.
Default	Whether this frame is default frame or not. The default frame is displayed first after the TouchPAD device turns on.
ProgramStyle	Standard C or Ladder

Event handlers of a frame

For example, we have a frame which is named “frame1”, and
When entering the frame1,

- **OnCreate**: TouchPAD executes this OnCreate event handler of frame1 first.
- **OnShow**: TouchPAD adds all the widgets used in the frame1 after OnCreate is executed. Then executes the OnShow. (So OnShow has widgets to use)

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

- **OnPaint**: whenever TouchPAD needs to paint its screen. OnPaint is executed after OnShow when TouchPAD just switches to the frame1.

When leaving the frame1,

- **OnHide**: TouchPAD executes OnHide first,
- **OnDestroy**: TouchPAD removes all the widgets used in the frame1 after OnHide is executed. Then executes the OnDestroy.

5.4.3. Rectangle

Unique Properties of Rectangle

properties	description
BrushStyle	The style used to fill to a rectangle

5.4.4. Ellipse

Unique Properties of Ellipse

properties	description
BrushStyle	The style used to fill to an ellipse

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.5. Text

Another way to put a Text (a string) on the frame

Simply copy an text from the clipboard and paste it on the frame design area of HMIWorks. HMIWorks then create a Text component and then load the string from clipboard automatically.

Unique Properties of Text

properties	description
Alignment	This property determines where to locate the string, Left, right, or center. (LeftJustify, RightJustify, or Center) Note: This property is enabled only when AutoSize=True
AutoScaleFontSize	Automatically scale the font size to fit the rectangle which encloses the Text. Note: This property is enabled only when AutoSize=True.
AutoSize	True or False. This property is used to indicate that whether the size of the rectangle which encloses Text can be automatically changed to cover the whole string.
BrushStyle	The style used to fill the rectangle that encloses the Text
TextAsImage	True or False. Whether the text is stored as an image or not. If the text is treated as an

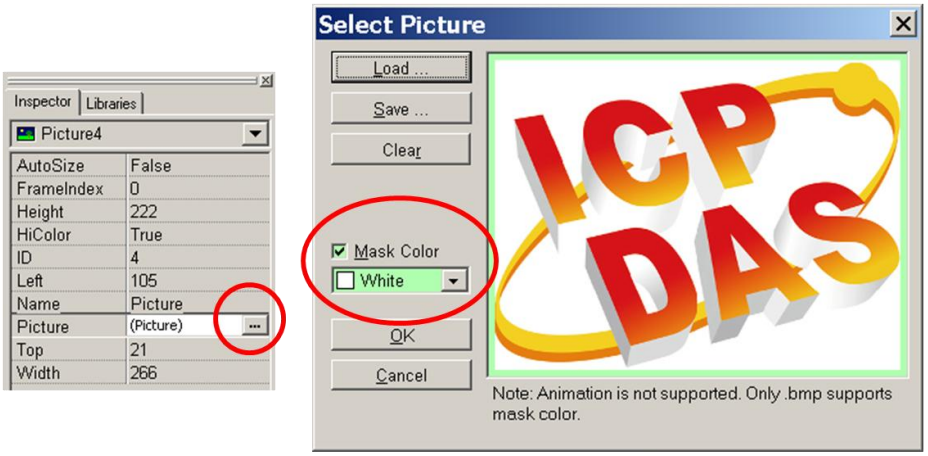
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

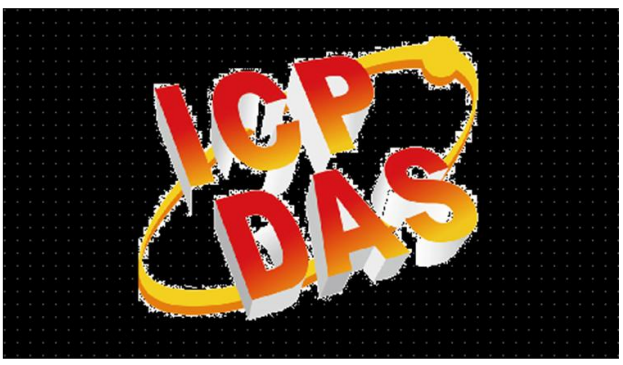
	image, it will take more space to store and more time to download.

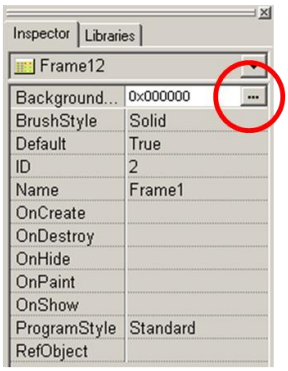
5.4.6. Picture

Loading a Picture

1. Click the “Picture” field in the **Inspector** (the “…” button) to open the “**Select Picture**” dialog to load a picture. There’s a “Mask Color” option to achieve transparency as shown below. For now, only .bmp files are supported for the “Mask Color” option.







As you can see above, we select the “Mask Color” option as white to

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

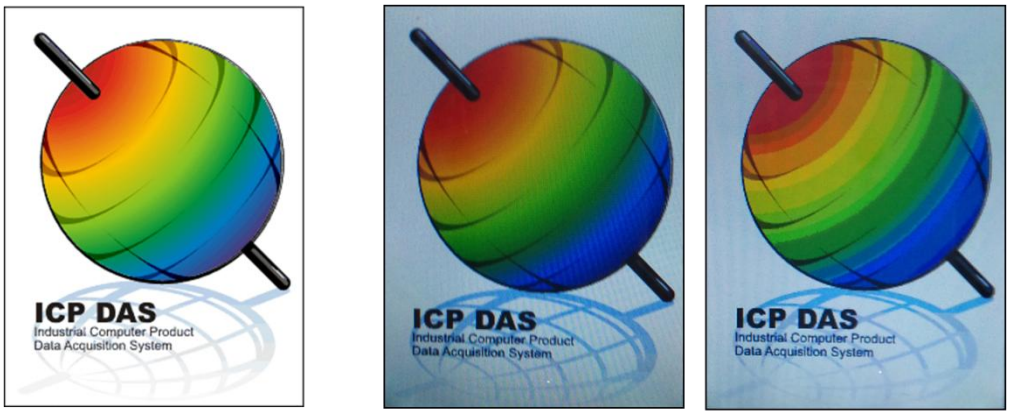
mask the white color, that is, the area of white color becomes transparent. Change the background color of the frame to black to illustrate the effect.

2. Or you can just copy an image from the clipboard and paste it on the frame design area of HMIWorks. HMIWorks create a Picture component and then load the image from clipboard automatically.

Unique Properties of Picture

properties	description
AutoSize	True or False. This property is used to indicate that whether the size of the Picture can be changed or not.
FrameIndex	Ignored
HiColor	True or False. This property decides whether the loaded picture is stored as 16-bit color (True) or 8-bit color (False). The default option is 8-bit color.
Picture	The picture to be loaded

Trade-off between firmware size and resolution



In HMIWorks | HiColor = True on TouchPAD (189KB) | HiColor = False on TouchPAD (69KB)

Above is the comparison between “HiColor = True” and “HiColor = False”. The left picture is original one in HMIWorks. The two right-side pictures are real photos. One is “HiColor = True” and the other is “HiColor = False”. As you can see, setting HiColor to False makes the photo have a

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

not-smooth gradient part while setting HiColor to True does not. Because 8-bit color does not have enough color (256 only) to represent the picture, similar colors are represented by the same color and this results in not-smooth gradient.
However, preventing pictures from not-smooth gradient costs TouchPAD bigger size of memory. Take above picture for example, setting HiColor to True uses memory of 189KB but setting HiColor to False costs only 69KB.

5.4.7. Line

Unique Properties of Line

properties	description
None	N/A

5.4.8. TextPushButton

What is a TextPushButton?

A TextPushButton is a button with a Text on it. When a TextPushButton is pressed and not released, the status is changed. But the status is restored back to the original state after you release it.

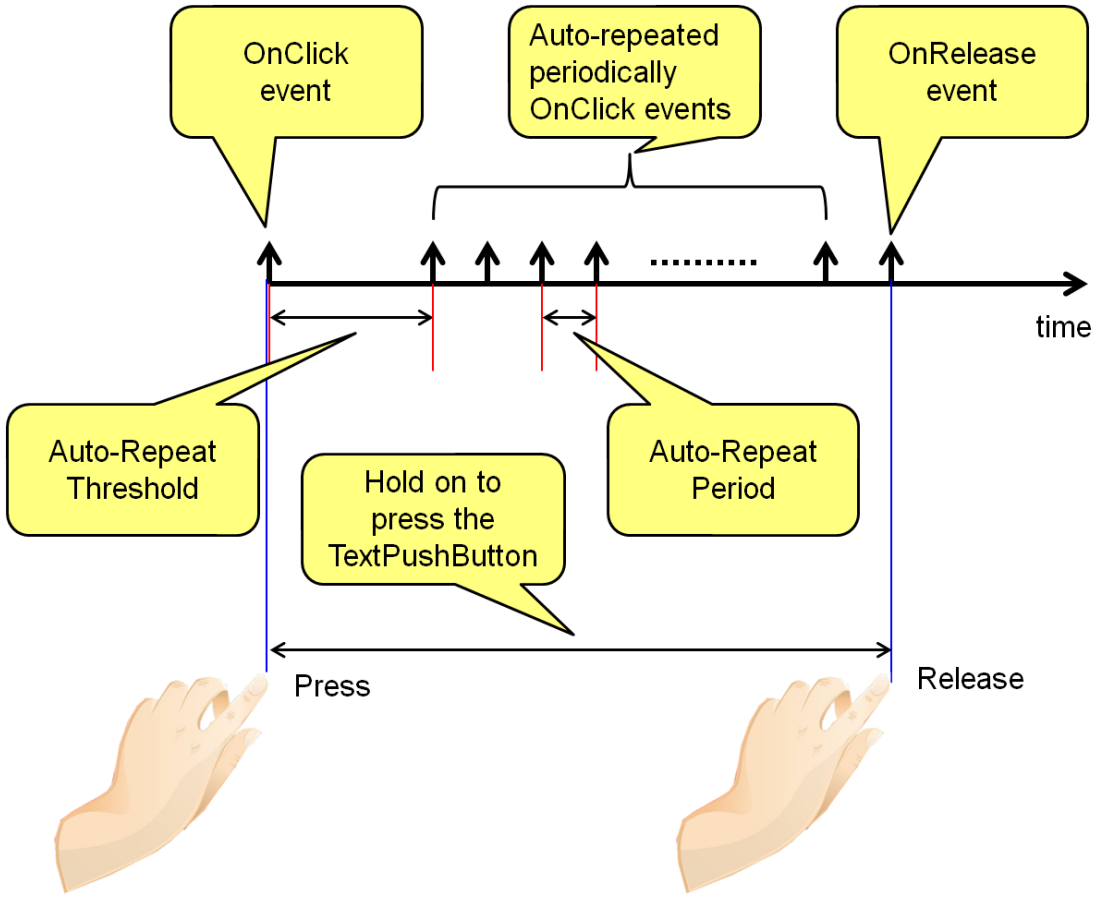
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Unique Properties of TextPushButton

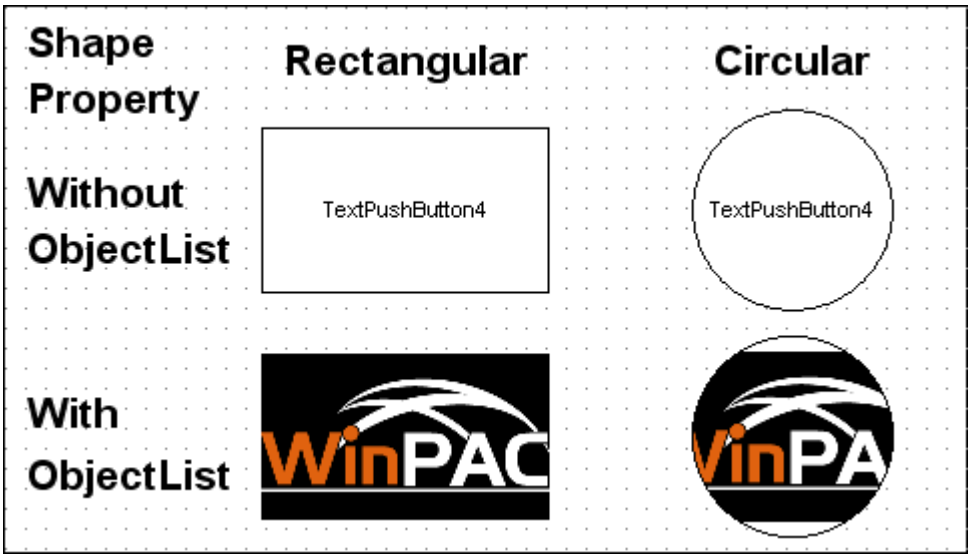
properties	Description
AutoRepeatPeriod	The period to hold on to press the TextPushButton to trigger one OnClick event again when in the programming type “Standard C”. (unit: ms)
AutoRepeatThreshold	After pressing the TextPushButton to trigger the OnClick event and continuing pressing, this property determines the threshold of the time that is required to trigger the first periodical OnClick event (not the first OnClick event) when in the programming type “Standard C”. (unit: ms)
PressFillColor	The color used to fill the TextPushButton when the TextPushButton is touched (but not yet released)
Shape	The shape of a TextPushButton, Circular or Rectangular.

Triggered events

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

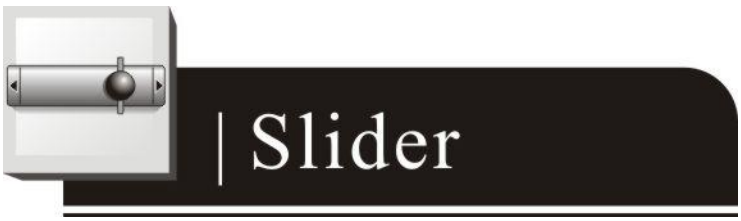


An example demonstrate the Shape property



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.9. Slider



What is a Slider?

A Slider is a control element used to set levels. Usually, a Slider is used in volume control.

Unique Properties of Slider

properties	description
BackgroundFillColor	The color used to fill the background of the Slider. The color is represented by a three byte value in the hexadecimal form. From the highest byte to the lowest, it is the blue byte, the green byte, the red byte in sequence.
BackgroundTextColor	The color of the text in the background of the Slider. The color is represented by a three byte value in the hexadecimal form. From the highest byte to the lowest, it is the blue byte, the green byte, the red byte in sequence.
Max	The maximum value of the Position
Min	The minimum value of the Position
Position	The value where the slider locate (between Max and Min)
Vertical	The direction of the Slider

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.10. BitButton

What is a BitButton?

A BitButton is a button with 3D appearance and the status rebounds back if releasing the pressed button. When you press it, you can see that the BitButton is pressed “down”. This 3D-like appearance is achieved by two images so that it takes more spaces to store and more time to download than a Text PushButton.

5.4.11. HotSpot

What is a HotSpot?

HotSpot decides an area which is capable of responding to on-click events. Usually, putting a HotSpot on the Drawing components (that is, Rectangles, Ellipses, Texts, Pictures, and Lines) makes them to respond to on-click events. After downloading to TouchPAD, a HotSpot is invisible.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.12. CheckBox

What is a CheckBox?

A CheckBox is a control element that provides a yes-no choice.

Unique Properties of CheckBox

properties	description
BoxSize	The size of the checking box
DisplayType	How to display the pictures which are loaded from RefObject property.
Selected	True or false. Whether the CheckBox is checked or not

5.4.13. Label

What is a Label?

A Label is a Text put on TouchPAD to give information that may change at the run time.

Unique Properties of Label

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

properties	description
Alignment	This property determines where to locate the string, left, right, or center. (LeftJustify, RightJustify, or Center)
DecimalDigits	The power to which ten must be raised to produce the value, say divisor, which is used to divide the value of the associated tag of this Label. The value of the tag must be divided by the divisor to show on the screen to represent decimal digits. **Note**: The property is supported only in programming type “Ladder”.

Representing decimals for Ladder Designer

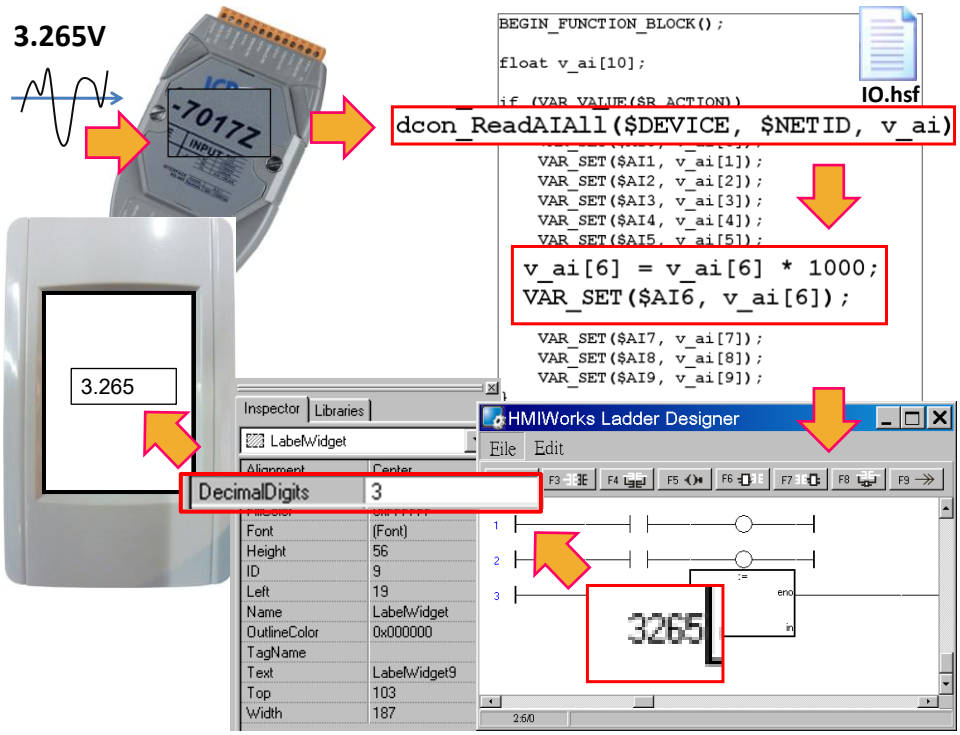
The numbers used in the **Ladder Desinger** are all integers. The decimals are not accepted in the **Ladder Designer**. However, in some cases, users may need to calculate or display decimals. So we provide a work-around method to handle these cases.

Take the I-7017Z module for example. Supposed that we use the I-7017Z module to read an analog value 3.265V back from the remote side and we want to display decimals on the TouchPAD devices. But the **Ladder Designer** supports only integers. So we must handle this drawback to directly read back the AI value from the I-7017Z module in the **Ladder Designer**.

1. Set the property “DecimalDigits” to the number of digits in the right of the decimal point. For example, we set DecimalDigits to 3.
2. Modify the I/O module’s IO.hsf. Let the read back AI value multiplied by ten of the n-th power where n is the value of “DecimalDigits”. You can find out I/O module’s IO.hsf file in the following locations: “[HMIWorks_install_path]\ bin\Modules\”.
   For example, IO.hsf of I-7017Z is located in “C:\ICPDAS\HMIWorks_Standard\bin\Modules\I-7000\I-7017Z”, where “C:\ICPDAS\HMIWorks_Standard\” is the installation path of HMIWorks. And we modify the IO.hsf to make v_ai[6] = v_ai[6] * 1000; Supposed we use channel 6 to read back AI value.

As shown in the figure below, you can see that the tag “$AI6” in the **Ladder Designer** is 1000 times of the real value. With DecimalDigits set to 3, the correct value 3.265 is displayed on TouchPAD.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Representing decimals in the C language

In the frame of “Standard C”, representing decimals may be difficult since “sprintf” function is not supported in HMIWorks.

We use “usprintf” (or “usnprintf”) to replace “sprintf”, but “usprintf” does not support the argument “%f”. In order to display a floating-point value, we provide a new API function for this purpose, the “FloatToStr” function as shown in the example below.

```
void TextPushButton4OnClick(tWidget *pWidget)
{
   float ret_sin;
   float angle = 1.57;
   static char str_sin[16];

   // sin
   ret_sin = sin(angle);

   // int FloatToStr(char *buf, float fVal, int precision);
   // the precision determine the number of the digits after the decimal point
```


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

```
    FloatToStr(str_sin, ret_sin, 3);
    LabelTextSet(&Label5, str_sin);    // The result is 1.000
}
```

5.4.14. RadioButton

What is a RadioButton?

The RadioButtons is used for a “one-of-many” selection. That is, only one of the RadioButtons in a particular group (we call it RadioGroup) can be selected.

Unique Properties of RadioButton

properties	description
CircleSize	The size of the checking circle
RadioGroup	The group in which only one RadioButtons can be selected. Each frame has at most 8 RadioGroups, from Group0 to Group7.
Selected	True or false. Whether the RadioButton is selected or not
SerialNumber	The unique number started from 0 which is used to identify a RadioButton in a particular RadioGroup. The SerialNumber property is used only for users to know about which RadioButton is to use, for example, when using the RadioButtonGroupValueSet function. **Note1**: this is a read-only property and is assigned automatically. **Note2**: when a RadioButton assigned a tag with the TagName property, then all the other RadioButtons in the same RadioGroup are assigned the same tag to their TagName property at the same time. Depending on the value of the tag (usually, the tag represent a I/O from the remote side), certain RadioButton is selected if its SerialNumber property is equal to the value of tag.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

TagName property has different behavior

Unlike other widgets, several RadioButtons in the same RadioGroup have the same TagName property. Since RadioButtons together provide a “one-of-many” selection, the value of the TagName property is the same among all the RadioButtons in a particular RadioGroup.
For example, supposed we have 3 RadioButtons, 0, 1, 2, where 0, 1, 2 are their SerialNumbers. And they are all specified in a RadioGroup, Group0. If we specified the TagName with an AI tag, named Dev_AI0, then we have the following behaviors:

1. When Dev_AI0 = 0, only RadioButton with SerialNumber 0 is selected (while the other two are unselected).
2. When Dev_AI0 = 1, only RadioButton with SerialNumber 1 is selected.
3. When Dev_AI0 = 2, only RadioButton with SerialNumber 2 is selected.

OnRadioChange property

Unlike the TagName property, each RadioButton has its own OnRadioChange event handler. An example as shown below:

```
void RadioButton6OnRadioChange(tWidget *pWidget, unsigned char
ucValue)
{
    //
    // ucValue is the serial number of the selected RadioButton in the
    // RadioGroup which contains this Radio Button. The RadioButton which
    // triggers this OnRadioChange event handler is not necessarily the
    // same as the selected RadioButton.
    //

}
```


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.15. Timer

Note

This component is supported only in programming type “Standard C”.

What is a Timer?

A Timer is a component that executes the OnExecute event handler every specified interval.

Using a Timer

Note that you should not worry about the size or the location of the Timer because the Timer is invisible when downloaded to the TouchPAD. Also it’s not necessary to put the Timer on the frame panel.

Unique Properties of Timer

properties	description
Enabled	Whether the Timer is enabled or not
Interval	The time span of two consecutive OnExecute events

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.4.16. PaintBox

Note

This component is supported only in programming type “Standard C”.

What is a PaintBox?

A PaintBox is a component which is used to paint shapes, such as rectangles, ellipses, etc., in the runtime.

Clearing a PaintBox

Use the “hmi_SetForeground” function to paint a white rectangle to clear the PaintBox as shown in the red box in the example below.
Refer to the API reference for more details.
ftp://ftp.icpdas.com/pub/cd/touchpad/document/english/api_reference/

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

```
int flag = 0;

void BitButton5OnClick(tWidget *pWidget)
{
  flag ++;
  WidgetPaint((tWidget*) &PaintBox4);
}


void PaintBox4OnPaint(tWidget *pWidget, tContext *pContext)
{
  if(flag % 2)
  {
    //blue; R-G-B
    hmi_SetForeground(pContext, 0x0000FF);

    hmi_FillRect(pContext,
                 WidgetLeft(pWidget) -10,
                 WidgetTop(pWidget) -10,
                 WidgetRight(pWidget) -10,
                 WidgetBottom(pWidget) -10
                 );
  }
  else
  {
    //white; R-G-B; used to clear the PaintBox
    hmi_SetForeground(pContext, 0xFFFFFF);

    hmi_FillRect(pContext,
                 WidgetLeft(pWidget),
                 WidgetTop(pWidget),
                 WidgetRight(pWidget),
                 WidgetBottom(pWidget));
  }
}
```

5.4.17. ObjectList

What is an ObjectList?

An ObjectList is a component which is used to maintain a list of library

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

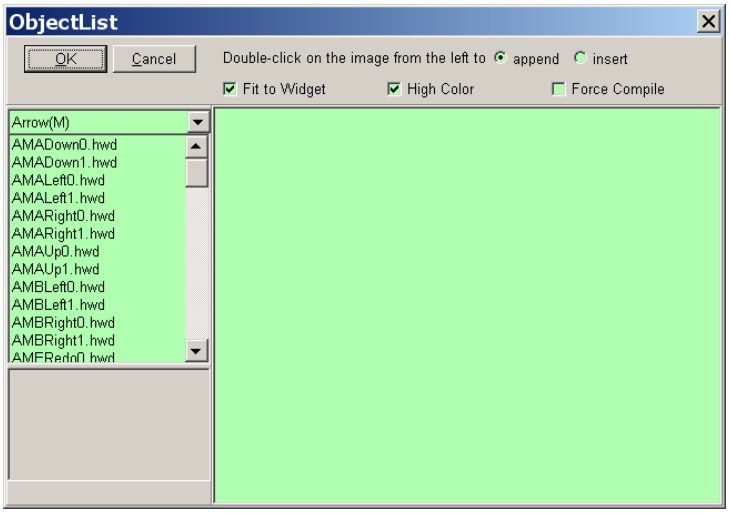
objects. Combined with “RefObject” properties of TextPushButton, Slider, CheckBox, and RadioButton components, users can easily toggle two or multiple images.

Using an ObjectList

1. Note that you should not worry about the size or the location of the ObjectList component because the ObjectList component is invisible after downloaded to the TouchPAD device.

2. The ObjectList component maintains a list of a library objects and is used in a component (e.g. CheckBox) with the RefObject property. After downloading to the TouchPAD device, the images of the library objects replace the original display of the component. When the state/value of the component changed, users see only the images of the library objects displays in the order in the ObjectList according to the state/value of the component.

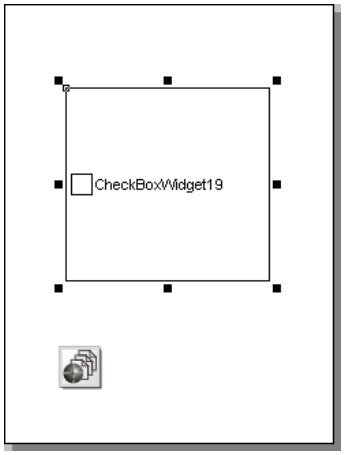
3. For example, add two library objects in the ObjectList by double clicking the ObjectList icon. Then the “ObjectList” window is displayed. Double click on the list of the library objects to add them to the right side pane.
   Note 1: To delete the library objects in the “ObjectList” window, double click on the objects in the right-side pane.



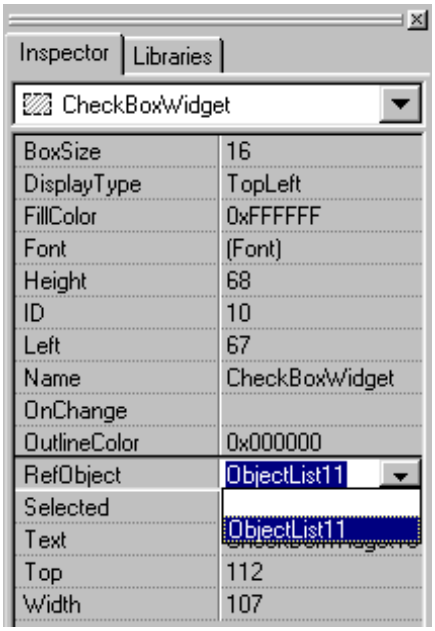
4. Click and drag a CheckBox component on the frame panel for example. Be sure to make the size of the CheckBox component

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

large enough to cover the whole image of the library object.



5. Go to the **Inspector** to select an option from the “RefObject” field for the CheckBox component. The selected ObjectList component is connected to the CheckBox component.



6. Build and download the project. You can see two images of the library objects toggle when the state the CheckBox component changes.

Options about images in ObjectList dialog

Options	Descriptions	Default
Fit to Widget	Resize the images in the ObjectList to cover the whole area of the widget which	True

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

	references to it.	
High Color	Render the images in the ObjectList as 16-bit color (high color) or 8-bit color when compiling.	True
Force Compile	Force HMIWorks to compile the images of this ObjectList which is not used by any widgets.	False

Note

To display transparent color (mask color) correctly, the following conditions must be satisfied.

1. The “Fit to Widget” option in the ObjectList dialog must be checked.
2. Each object of the ObjectList must contain only one Picture component. (Note that when you “add to library” the picture, it is grouped.)
3. TextPushButton with an ObjectList assigned to its RefObject property does not support the transparent (mask) color function when its Shape property set to “Circular”.

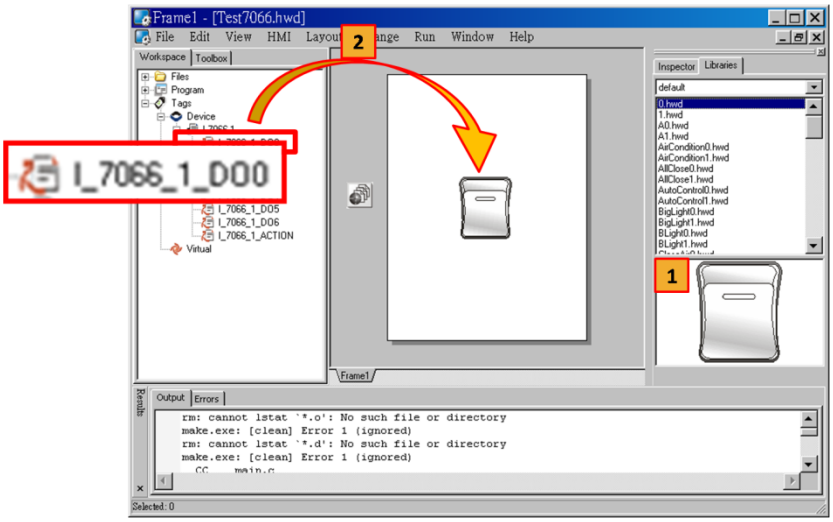
Unique Properties of ObjectList

Property	Description
Objects	The maintained library objects

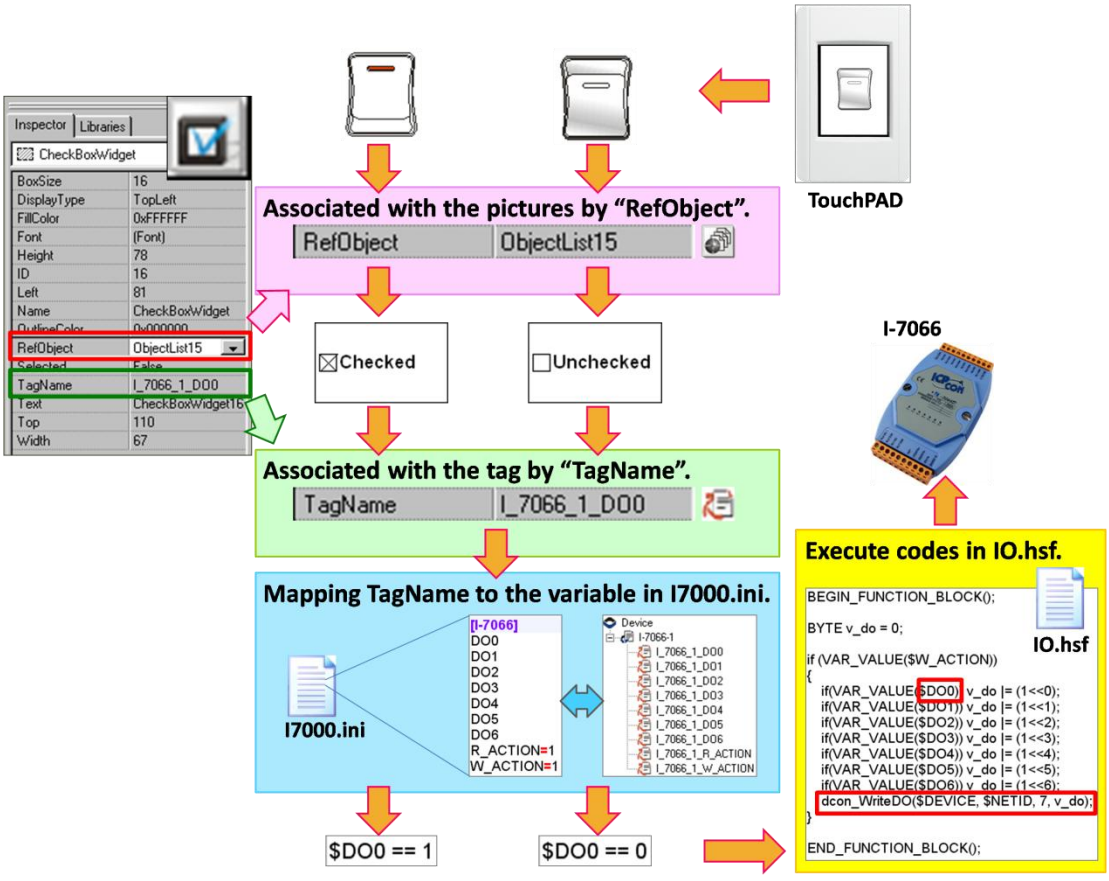
Relationships between TouchPAD and I/O module

Take the I-7066 module for example, click on the “**Register Devices**” option from the “**HMI**” menu or press **F3** on your keyboard to automatically generate tags and then drag and drop the tag on the frame.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



HMIWorks does the followings to build the relationships between the TouchPAD device and I/O modules.



Note: The TagName property takes effect only in the programming type Ladder. (It's easier in programming type "Standard C". Control the I/O by using API function, dcon_WriteDO, in the event handler of the CheckBox.)

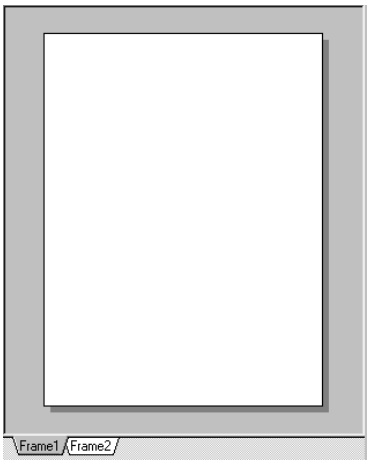
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.5. Menus

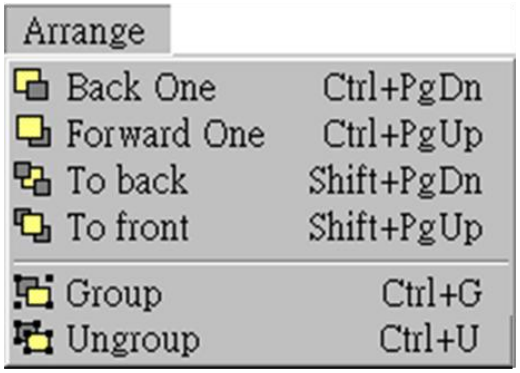
All the menus can be accessed from menu bar or the popup menu.

The menu bar:

Right click on the frame design area, a popup menu is displayed.
The frame design area:

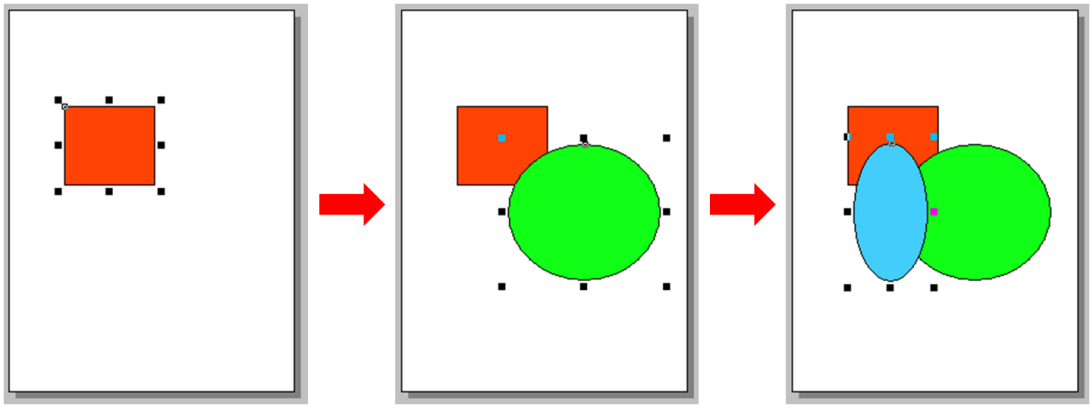


5.5.1. Cascading and Grouping, Arrange Menu



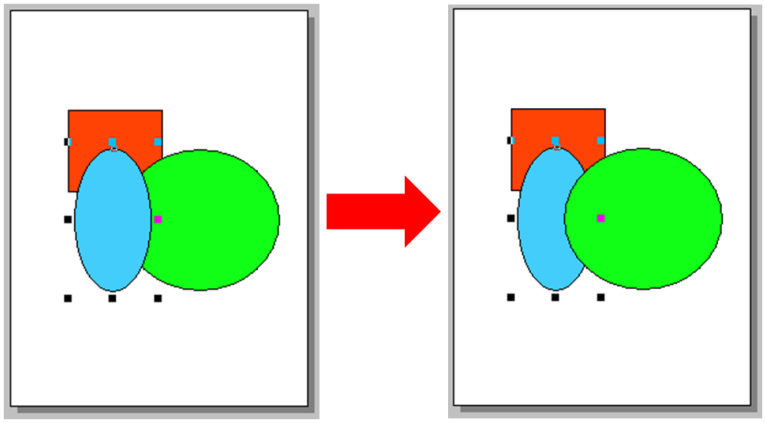
To demonstrate functions of cascading and grouping, first draw three shapes as followings:

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Back One

Make the selected object go down a level of the stacks.
For example, select the blue ellipse and click the “**Back One**” option in the “**Arrange**” menu. You can see that the blue ellipse goes down one level in the stack.



Forward One

Repeat the similar procedure for the “**Forward One**” option from the “**Arrange**” menu to go up a level in the stack.

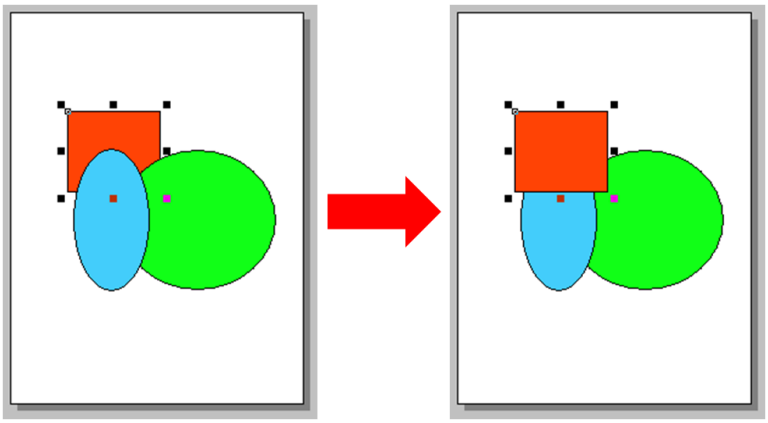
To back

Repeat the similar procedure for the “**To back**” option from the “**Arrange**” menu to make the selected object go down to the lowest level of the stack.

To front

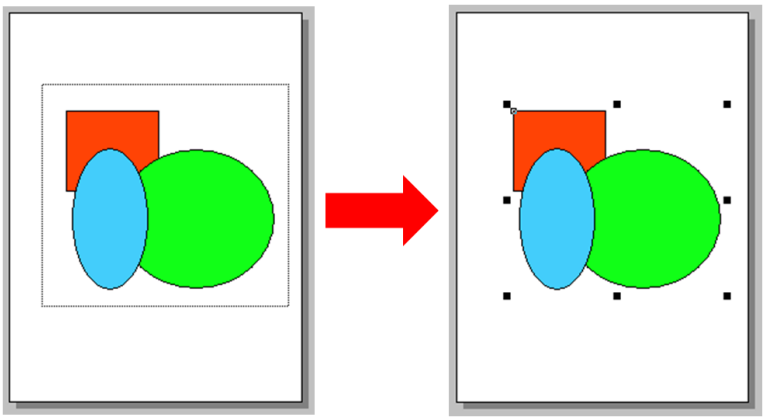
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Repeat the similar procedure for the “**To front**” option from the “**Arrange**” menu to make the selected object go up to the highest level of the stack.
For example, select the red square and click “**To front**” in the menu. You can see that the red square goes up to the highest level in the stack.



Group

Put components (the Drawing, the Widget and the System components) together as a set, that is, a group.
For example, first circle the items together by a mouse, and then click “**Group**” in the menu. You can see that they are grouped together.

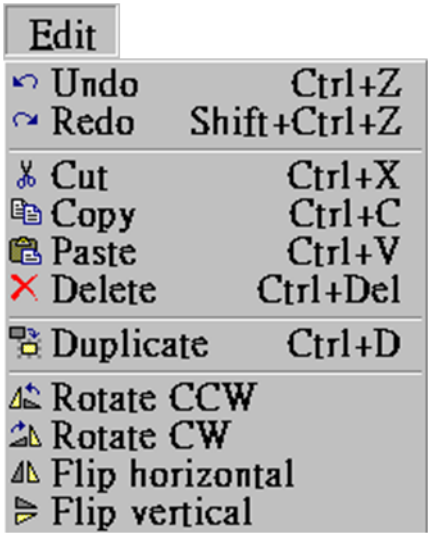


Ungroup

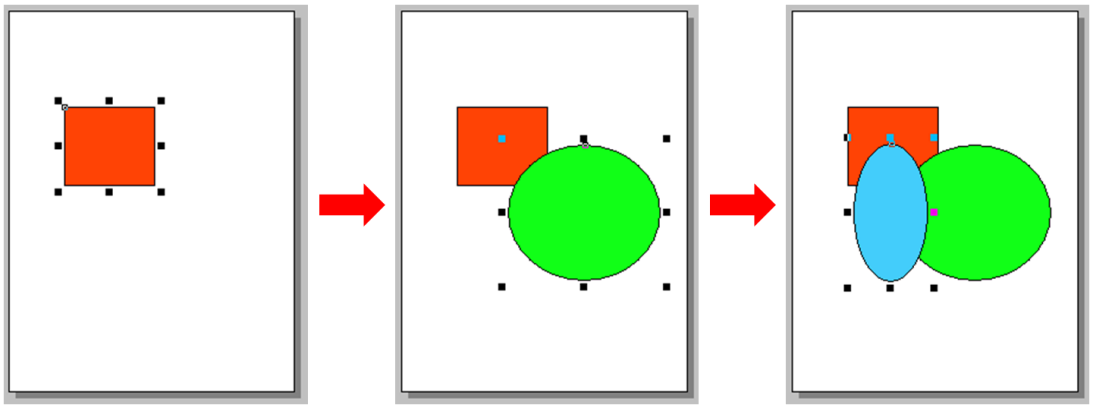
Break a group back into its original separate state.
For example, select the group and then click the “**Ungroup**” option from the “**Arrange**” menu.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.5.2. Rotating and Flipping, Edit Menu



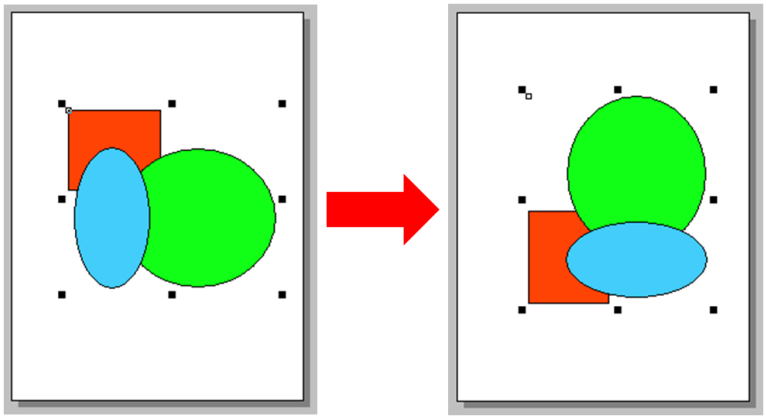
To demonstrate functions of rotating and flipping, first draw three shapes as followings:



Rotate CCW

Rotate the selected item in the counter-clockwise direction.
For example, first put three items into one group, select the group and then click the "**Rotate CCW**" option from the "**Edit**" menu. You can see that this group of shapes is rotated counter-clockwise.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

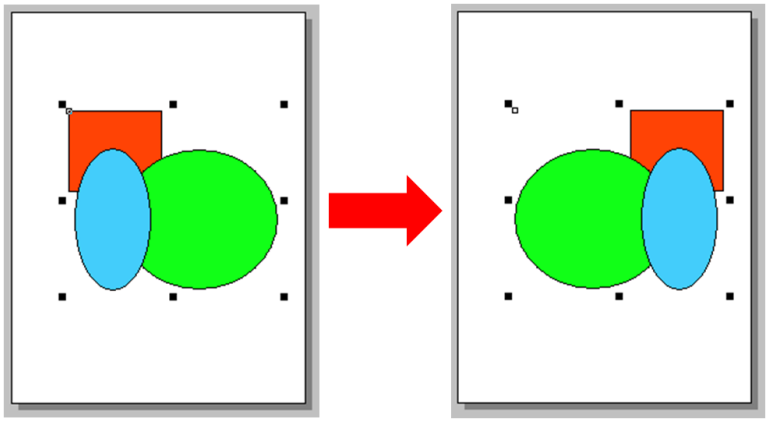


Rotate CW

Similar to the above step, click the “**Rotate CW**” option from the “**Edit**” menu to rotate the selected item in the clockwise direction
For example, first put three items into one group, select the group and then. You can see that this group of shapes is rotated clockwise.

Flip horizontal

Flip the selected item in the horizontal direction.
For example, first put three items into one group, select the group and then click the “**Flip horizontal**” option from the “**Edit**” menu. You can see that this group of shapes is flipped horizontally.



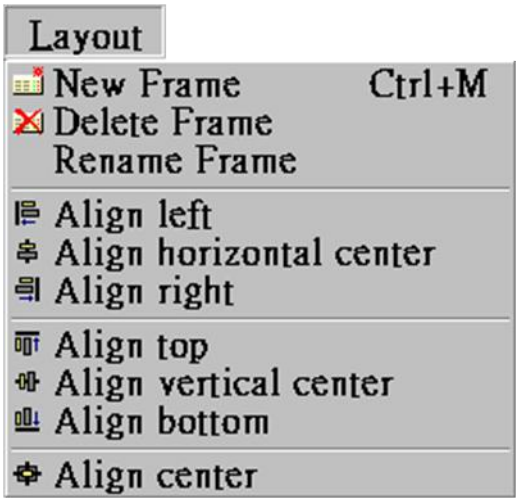
Flip vertical

Similar to the above step, click the “**Flip vertical**” option from the “**Edit**” menu to flip the selected item in the vertical direction.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

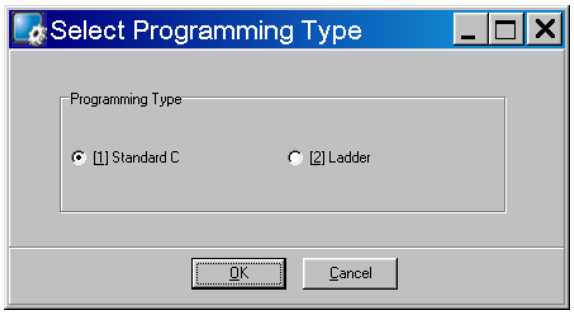
5.5.3. Frame Managing and Aligning, Layout

Menu



● Frame Management:

New Frame
Create a new frame
(select the programming type)



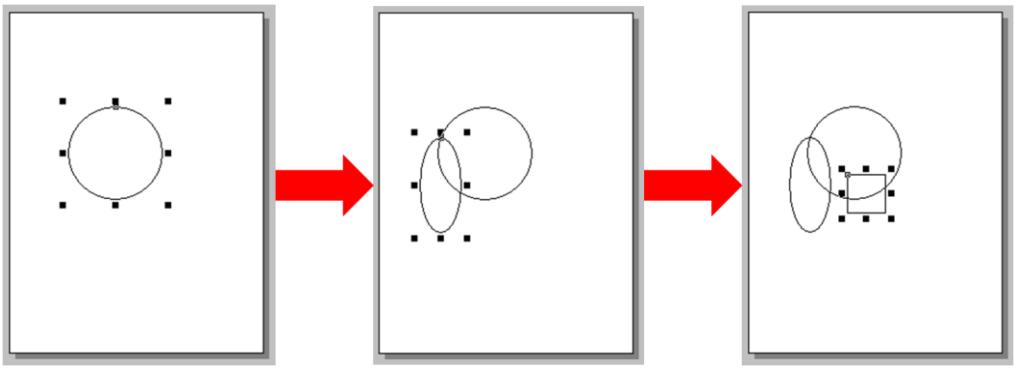
Delete Frame
Delete the current frame

Rename Frame
Rename a frame

● Alignment:

To demonstrate the functions of alignment, draw three shapes as followings

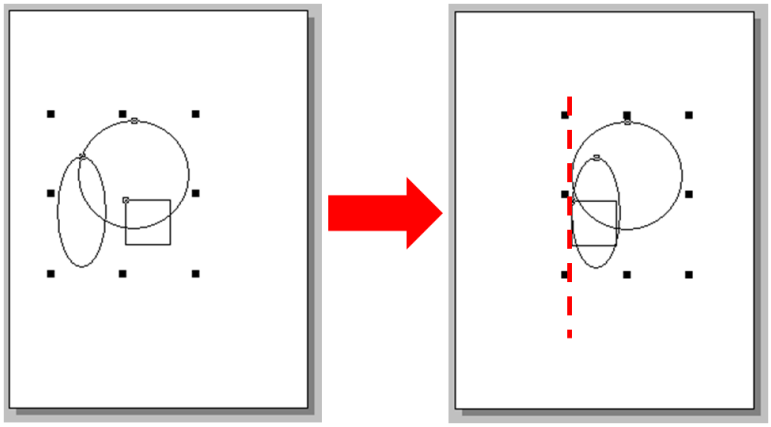
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Note: all alignment functions refer to the last shape you draw. In above example, all alignment functions refer to the square.

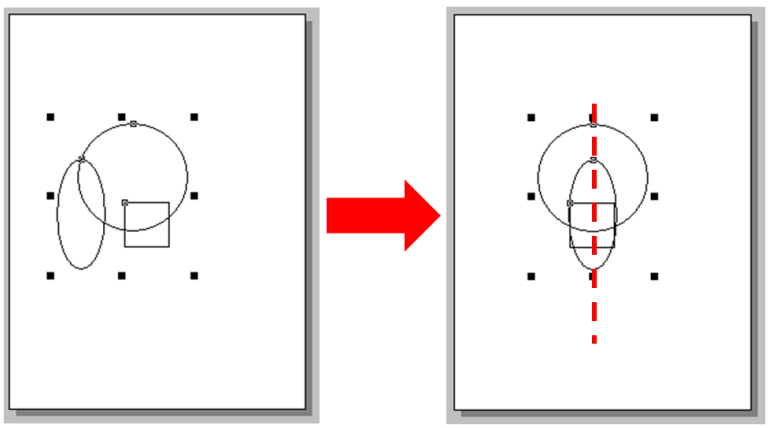
Align left

Align the leftmost edge of all the selected items to that of last-drawn item. For example, select all the items and then click "**Align left**" in the menu.



Align horizontal center

Align the horizontal center of all the selected items to that of last-drawn item. For example, select all the items and then click "**Align horizontal center**" in the menu.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

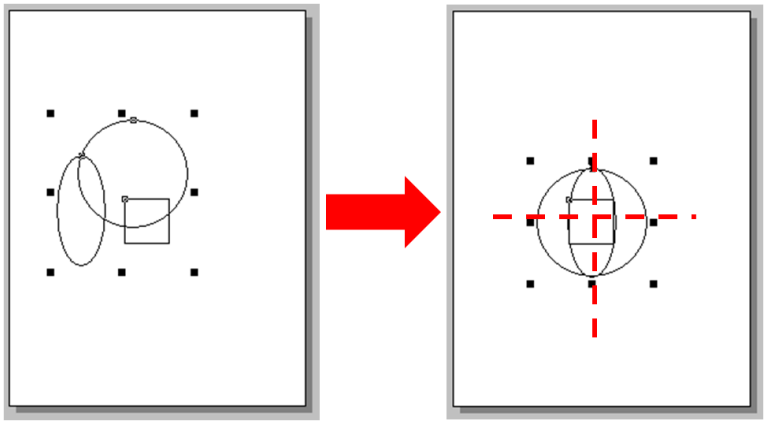
Align right
Align the rightmost edge of all the selected items to that of last-drawn item.

Align top
Align the topmost edge of all the selected items to that of last-drawn item.

Align vertical center
Align the vertical center of all the selected items to that of last-drawn item.

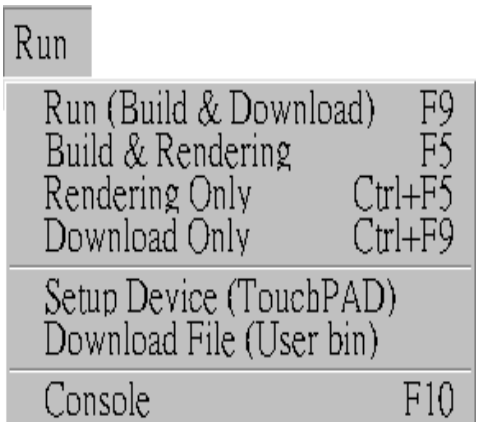
Align bottom
Align the bottommost edge of all the selected items to that of last-drawn item.

Align center
Align the center point of all the selected items to that of last-drawn item.
For example, select all the items and then click “**Align center**” in the menu.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.5.4. Build and Download to Run, Run Menu



Refer to the chapter, "Setup Devices and Connect to I/O", for "Setup Device".

Other Items in the Run Menu

Run (F9)

➔ Rendering + Build + Download

Rendering and build (F5)

➔ Rendering + Build (Compile and Link)

Rendering Only (Ctrl + F5)

➔ Generate source codes for frames, tools, ladders, etc.

Download Only (Ctrl + F9)

➔ Download the project's bin file to the TouchPAD devices

Download File (User bin)

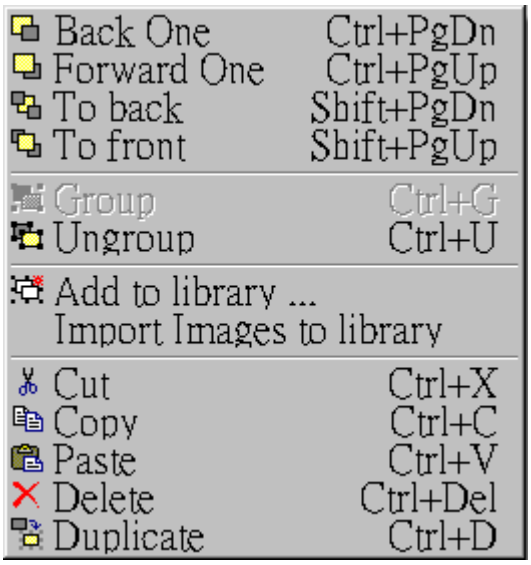
➔ Download a bin file to the TouchPAD devices

Console (F10)

➔ Open a console window (cmd.exe) with environment variables for HMIWorks. Users can modify the generated codes and then re-make the codes. (Use make.exe.)

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

5.5.5. Library Management, Popup Menu

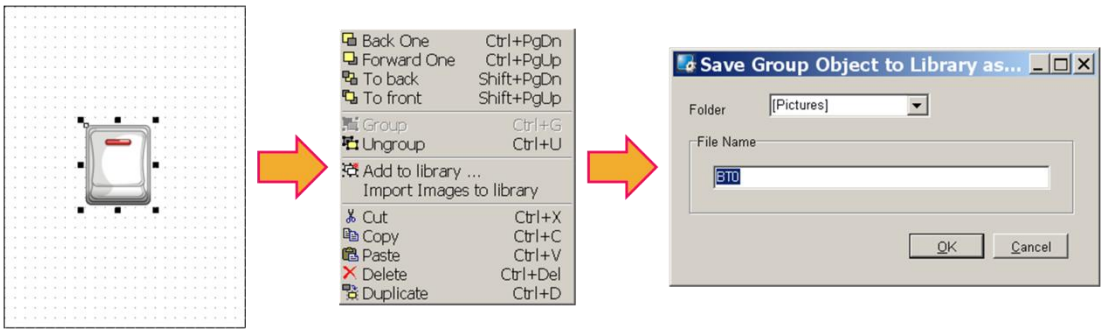


Adding items to library

All the items added have the file extension “hwd”.

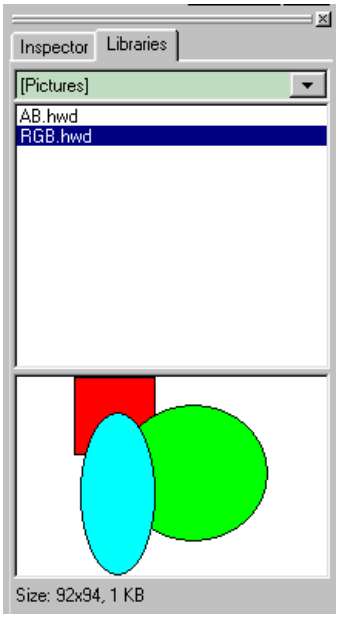
For example as described below:

1. Group the selected items if necessary.
2. Right click on the object we want to add to open the popup menu.
3. Click on “**Add to library …**”
4. Specify the folder we want the added object locates in the drop-down menu. The default is [Pictures].
5. Specify the name of the added object and save it to the library.



Note: you can preview the library object in the **library** window and the “size” information of that library object is shown below.

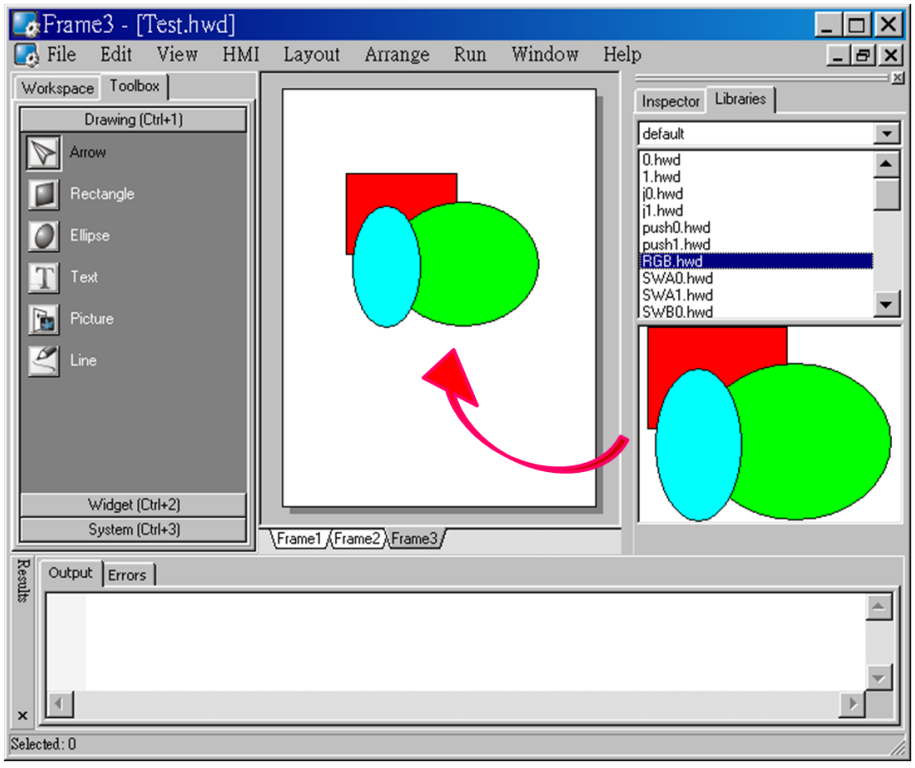
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Using items from library

For example as below:

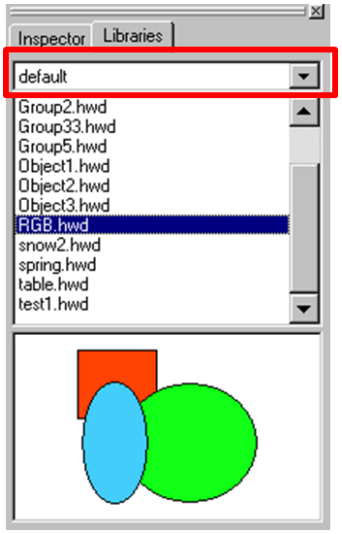
1. Click on the tab of "**Libraries**" to show the library panel.
2. Pick the object you want. You can preview the object in the preview box below.
3. Click (and not released) on the item in the preview box (or in the list) and then drag the item and drop it on the frame design area.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

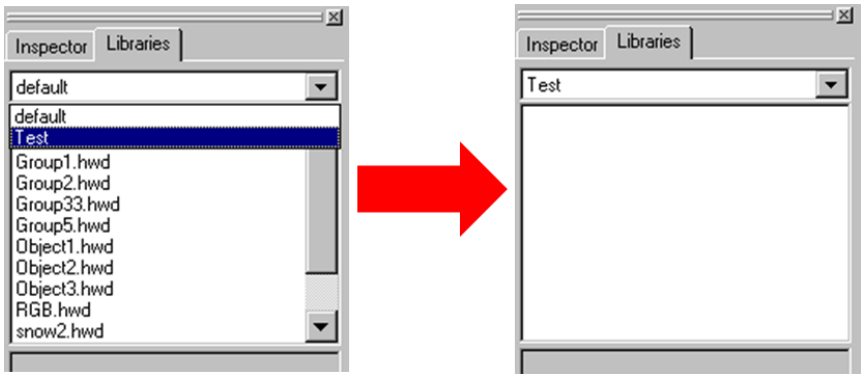
Adding a new folder into library window

Select the folder of the libraries as shown in the picture below.



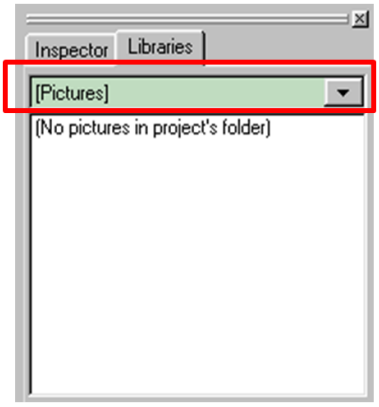
To add a new folder into the **library** window, create a new folder in the following path:
"HMIWorks_install_path\bin\Lib\" where the HMIWorks_install_path is the installation path of HMIWorks.

Supposed the installation path of HMIWorks is "C:\ICPDAS\HMIWorks_Standard". And we want to add a new folder named "Test" into the **library** window. Then all we have the do is creating a new folder named Test in the directory of "C:\ICPDAS\HMIWorks_Standard\bin\Lib".
And then re-open the **library** window, you can see that the new folder "Test" as shown below.
Of course, there's no library item in it. You should add items yourself.

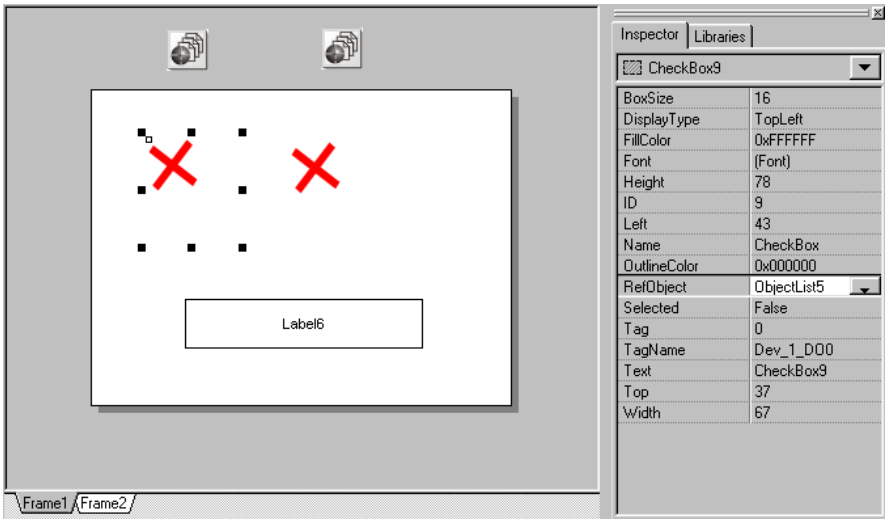


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

The special [Picture] directory in the project directory



1. Click the “**libraries**” tab, select the “[Picture]” directory from the dropdown menu as shown in the picture above.
2. Unlike others options in that dropdown menu, “[Picture]” directory is at the location of the project directory. Any library that is added to the “[Picture]” directory is always together with the project and makes the project portable among different computers.
3. When opening a project, a red cross will be shown on the frame panel if HMIWorks fails to load the image as shown in the below picture.



Import Images to Library

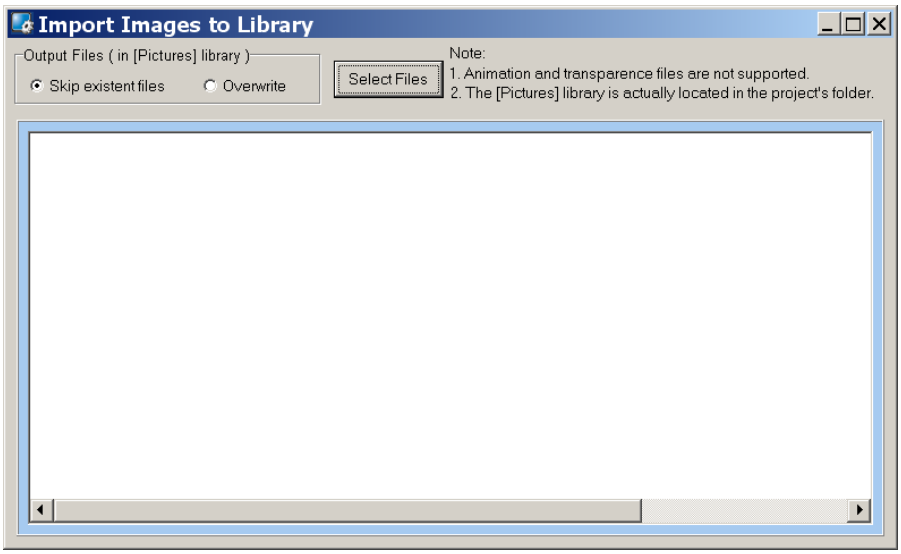
This option can also be accessed from the **File** menu.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Click this option to select more than one image files, transform them into the .hwd file format which HMIWorks can recognize and finally put these files in the [Pictures] folder in the current project directory.

Since the transformed .hwd files are put in the [Pictures] folder of a project, users should create or open a project to execute this option.

As shown below, click the "**Select files**" button to execute.


Import Images to Library
Output Files (in [Pictures] library)
Skip existent files
Overwrite
Select Files
Note:
1. Animation and transparence files are not supported.
2. The [Pictures] library is actually located in the project's folder.

Note: Now, we support JPG/BMP/WMF/EMF image formats.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

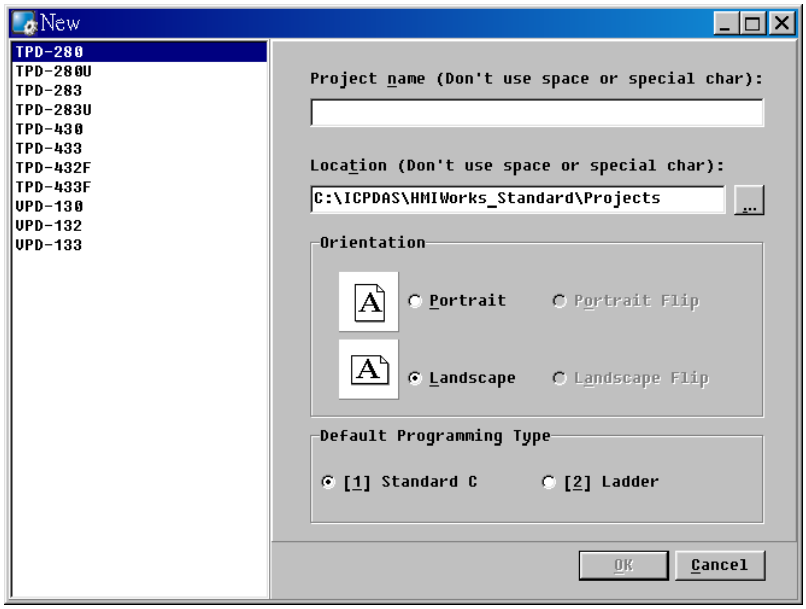
6. Making a Simple Project

There are two programming types in HMIWorks. In this chapter, we introduce how to build your first project for each programming type.

6.1. Your First Project Using Standard C

1. Creating a new project

Click the "**New…**" option from the "**File**" menu and then select the Model, specify the Project name, the Location, the Orientation, and the Programming Type.
Here we choose **programming type** as [1] Standard C.

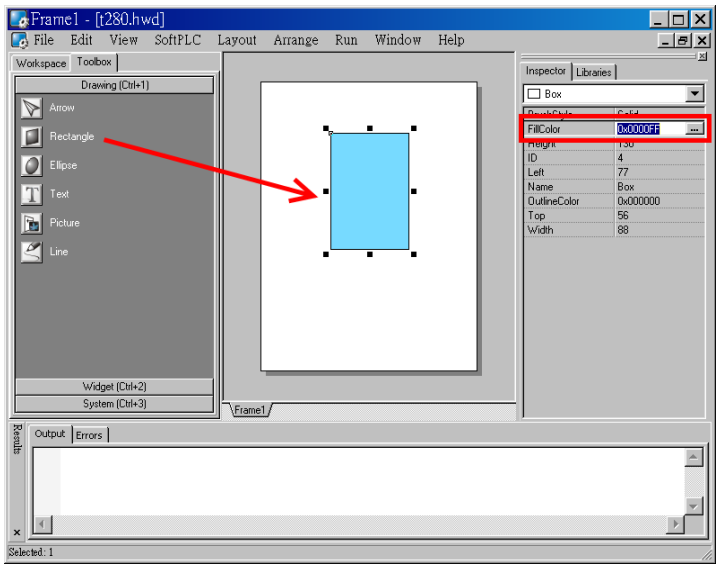


2. Designing the Graphic User Interface

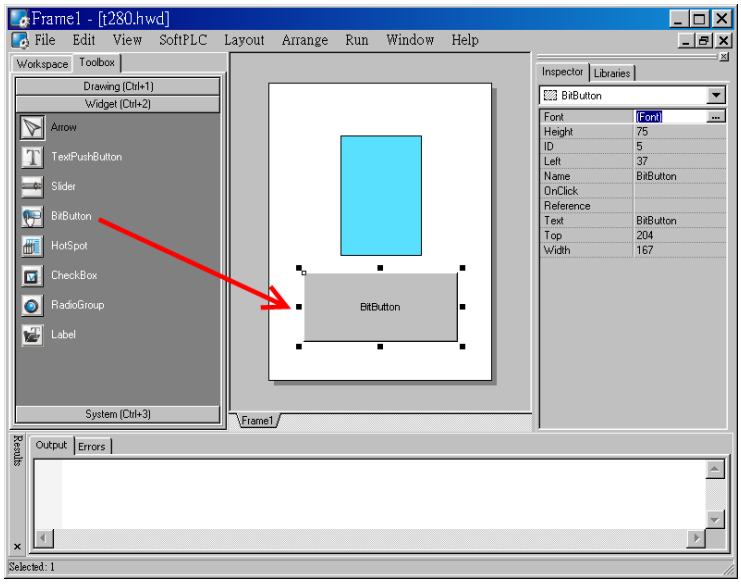
For example, draw a rectangular and fill the color. Of course, you can draw more complex and beautiful figures. Here, we simply demonstrate how to

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

make a simple project.



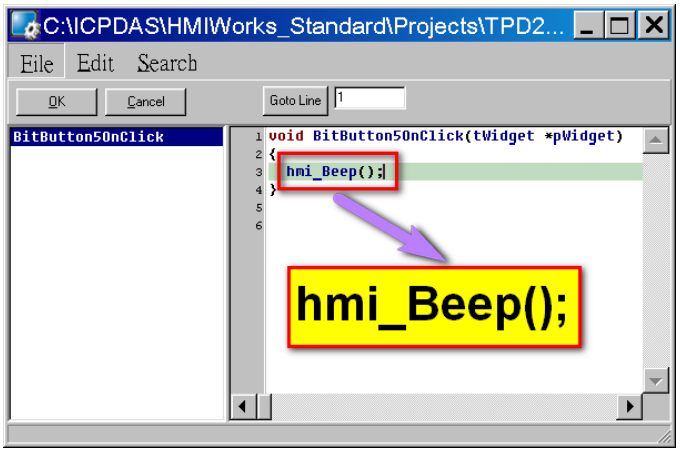
And then select a Widget. For example, pick a BitButton.



3. Modifying Source Codes

Double click the BitButton in the frame design area to open the programming window. Use “hmi_Beep();” to sound a beep for example, then press **OK**.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



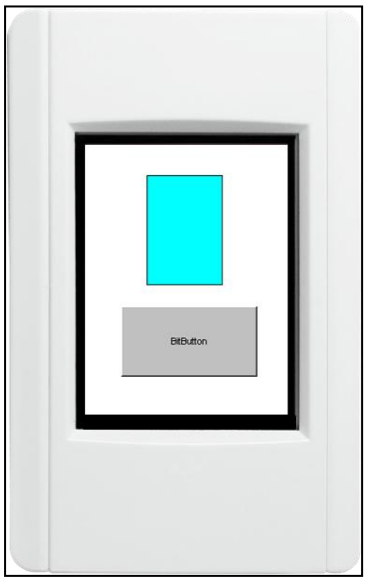
4. Setup Device

Refer to the "Setup Devices" section for details.

5. Compiling and Downloading to Run

After connecting to the TouchPAD device, press **F9** to run (or click the "**Run**" option from the "**Run**" menu.

As the shown in the figure below, pressing the button makes TouchPAD sound a beep.

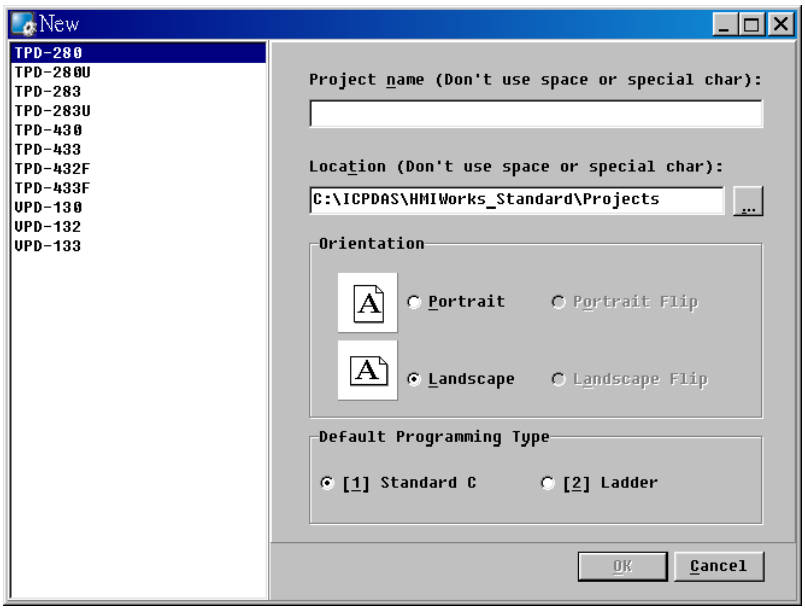


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

6.2. Your First Project Using Ladder

1. Creating a new project

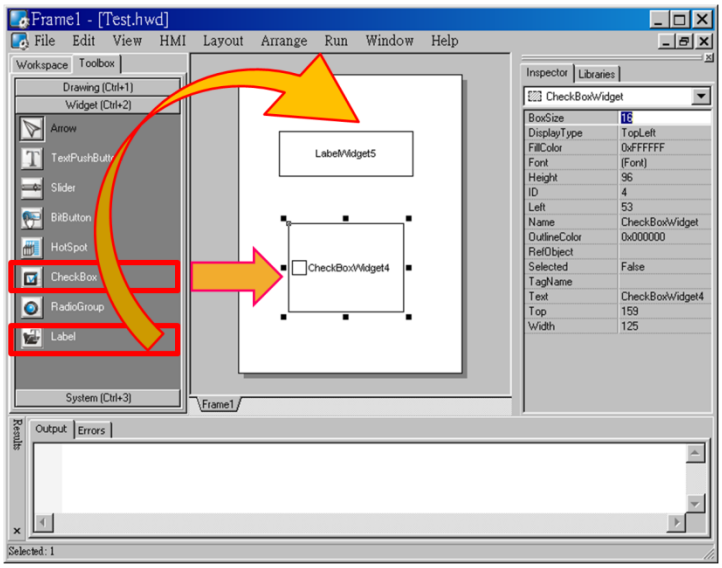
Click the “**New…**” option from the “**File**” menu, and then select the Model, specify the Project name, the Location, the Orientation, and the Programming Type. Here we choose **programming type** as [2] Ladder.



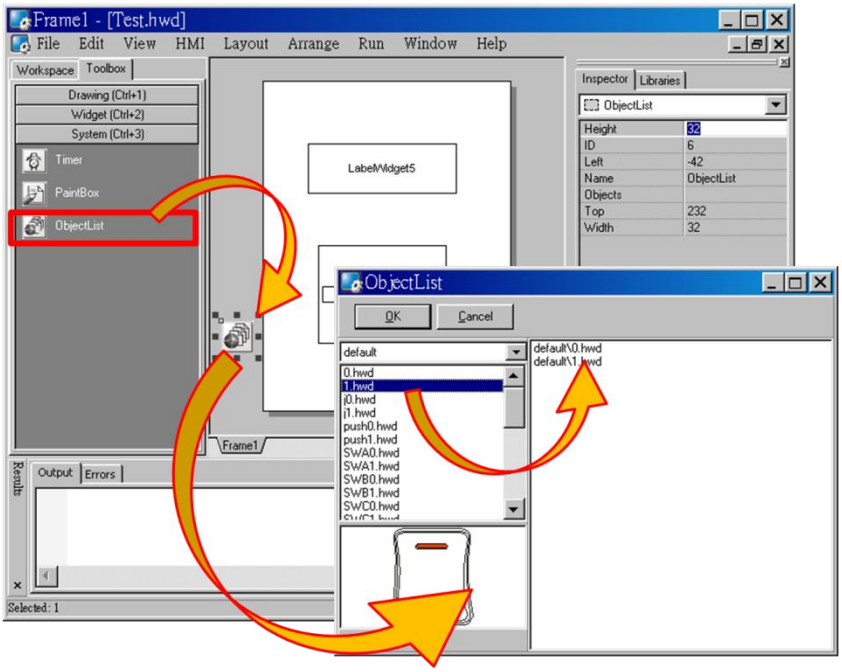
2. Designing the Graphic User Interface

For example, place a CheckBox component and a Label component on the frame panel. Here, we plan to take the CheckBox component as an input and the Label component as an output.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

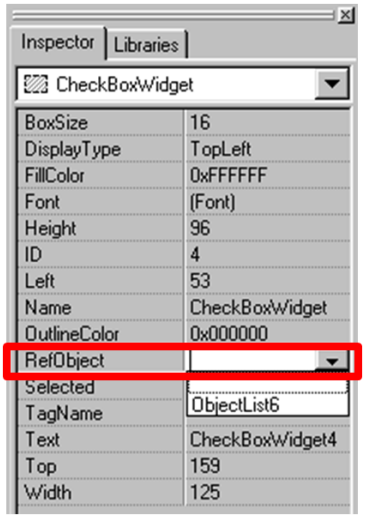


Select an ObjectList component and click on the frame design area. Double click the ObjectList icon to open the "**ObjectList**" window. In the "**ObjectList**" window, double click to select the pictures you want. Users need to double click on two pictures, one is for the checked state of the CheckBox component and the other is for the unchecked state. Press **OK** to finish this step.



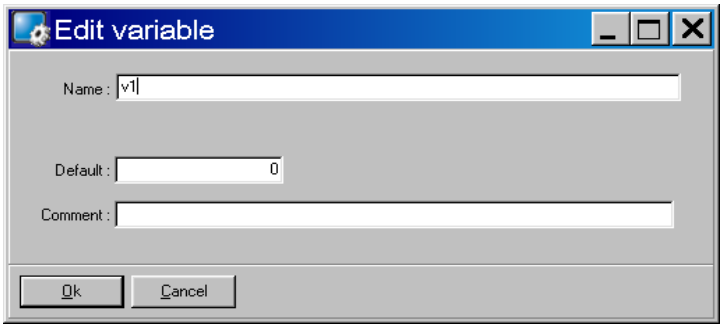
Make the CheckBox component refer to the ObjectList component by setting the property "RefObject" to the ObjectList component. Now toggling the states of the CheckBox component becomes the switching of the pictures in the ObjectList component.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

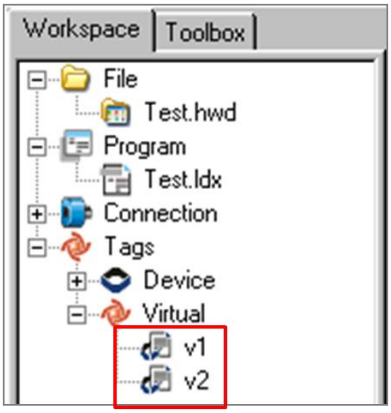


3. Designing the Ladder Diagram

First, add virtual tags (variables) for the ladder diagram. Press **F2** on your keyboard or click the "**New Virtual Tag**" option from the "**HMI**" menu. Here, we add two tags, v1 and v2, for example.



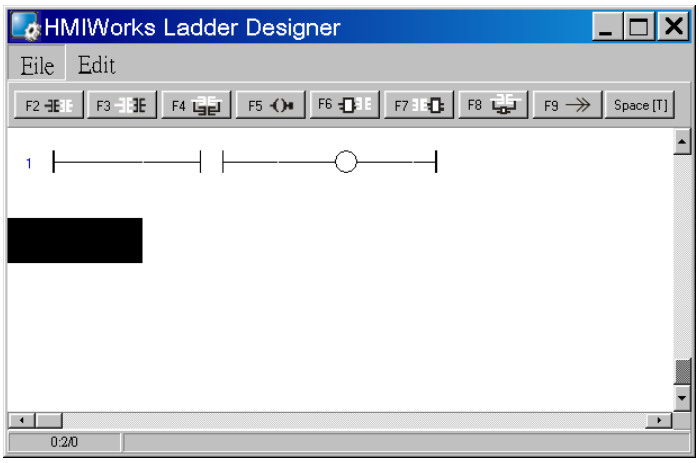
After adding the tags, users can verify in the **Workspace**.



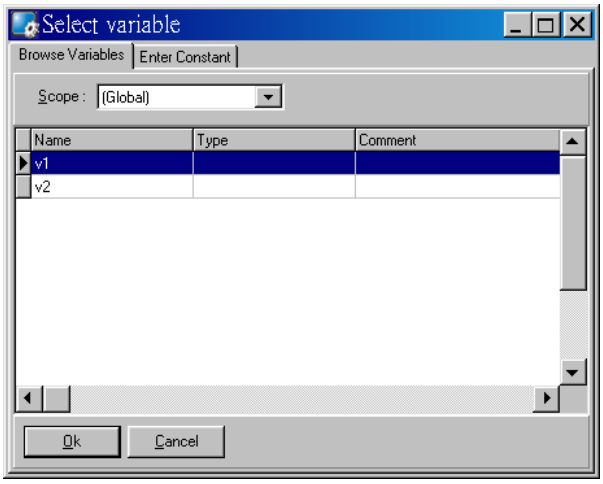
Press **F4** on your keyboard or click the "**Ladder Designer**" option from the "**HMI**" menu to open the **Ladder Designer** window. In the **Ladder**

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Designer window, press **F2** to create a new rung.

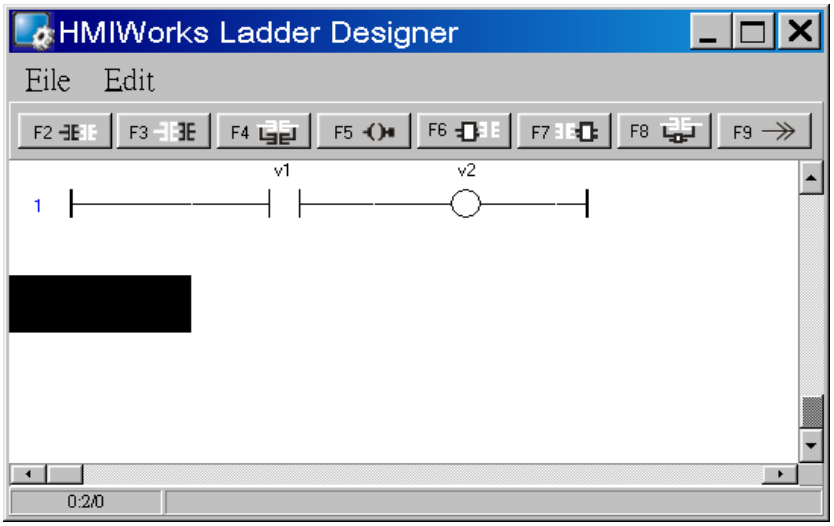


Double click the contact input of the first rung in the **Ladder Designer** window. Then the "**Select variable**" dialog box is displayed. Choose the variable to associate with the contact input.

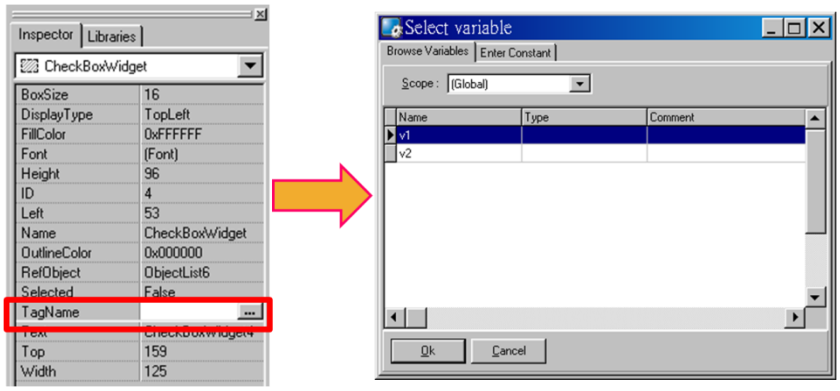


Here, we select variable v1 to associate the contact input. Repeat the same procedure to associate v2 with the coil output.

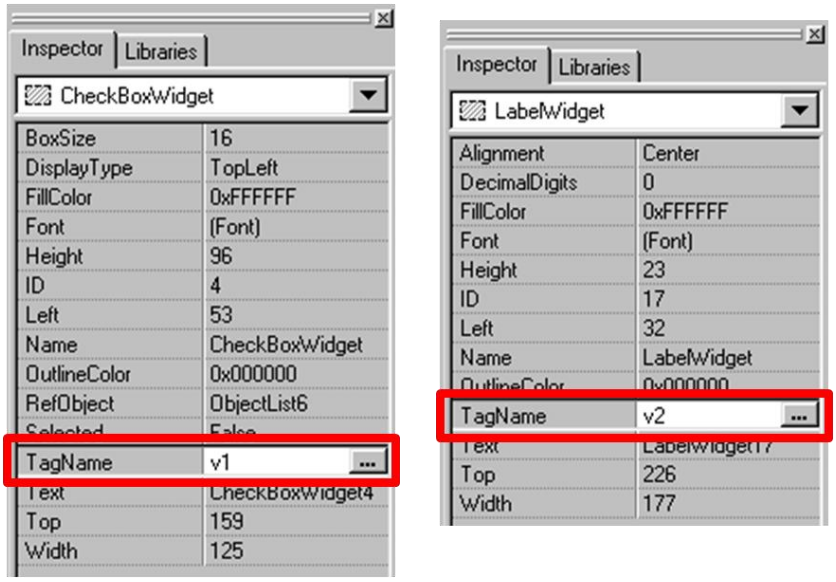
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



Then we associate the CheckBox component with the v1 tag and the Label component with the v2 tag by the “TagName” properties of themselves.



After setting the “TagName” properties, users can verify in the **Inspector**.



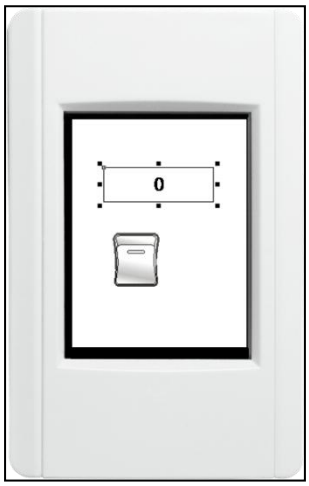
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4. Setup Device

Refer to the section “Setup Devices” for details.

5. Compiling and Downloading to Run

After connecting to the TouchPAD device, press **F9** on your keyboard to run (or click the “**Run**” option from the “**Run**” menu).
As shown in the figure below, pressing the button switches the value of the Label from 0 → 1, or 1 → 0.



6.3. Integrating TPD-280 Series with I/O modules

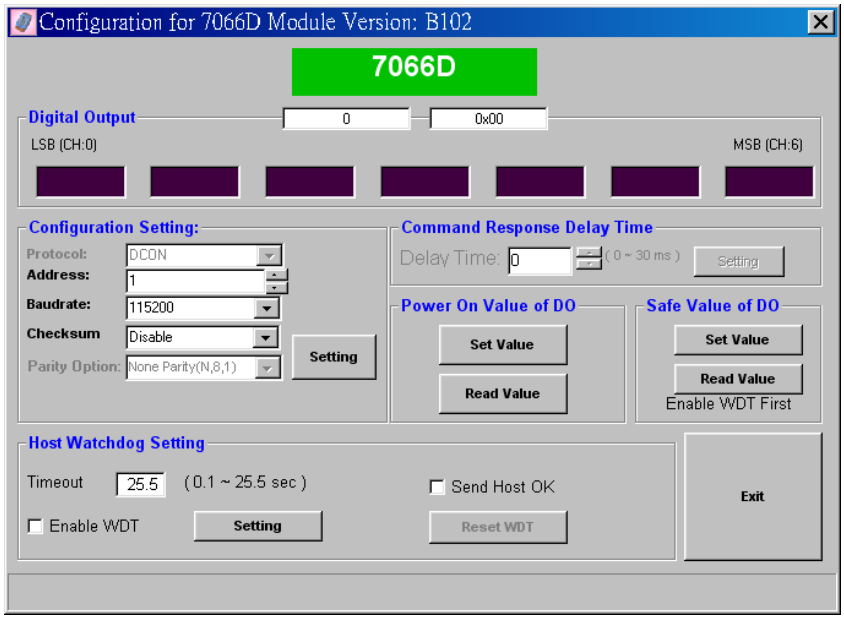
In this example, we use the TPD-280 device to control an I-7066 module, the 7-channel photo-MOS relay output module of ICP DAS. First, put the I-7066 module in the same RS-485 network of the TPD-280 device and configure the settings of the I-7066 module with the DCON Utility (baudrate, data bit, parity, stop bit, Net ID, etc.).

1. Using DCON Utility to Set Up I-7066

Download the DCON Utility to install and refer to its user manual.
ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/

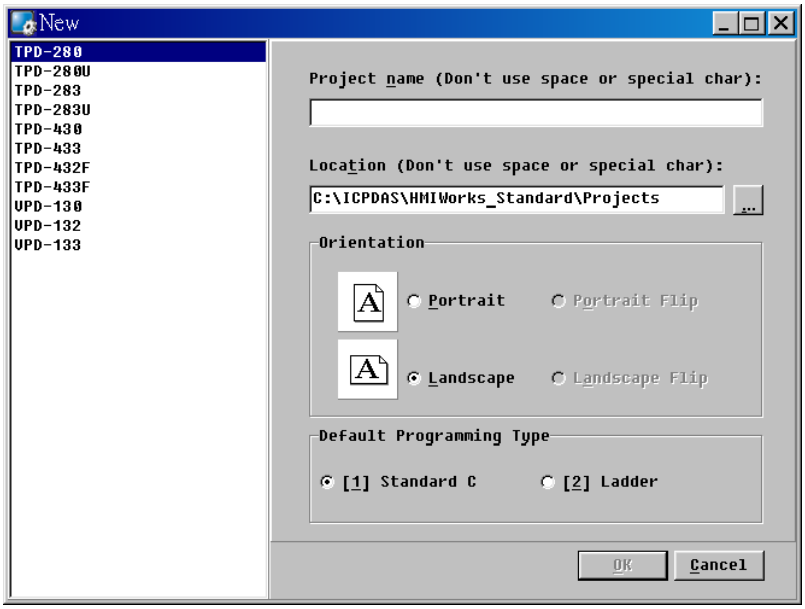
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Then use the DCON Utility to set up the I-7066 module. At least set the followings: Address(Net ID), Baudrate, Databit, Parity, and Stopbit.



2. Creating a new project

Click the "**New…**" option from the "**File**" menu and then select the Model, specify the Project name, the Location, the Orientation, and the Programming Type. Here we choose **programming type** as [2] Ladder.



3. Designing the Graphic User Interface

We can skip this step.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

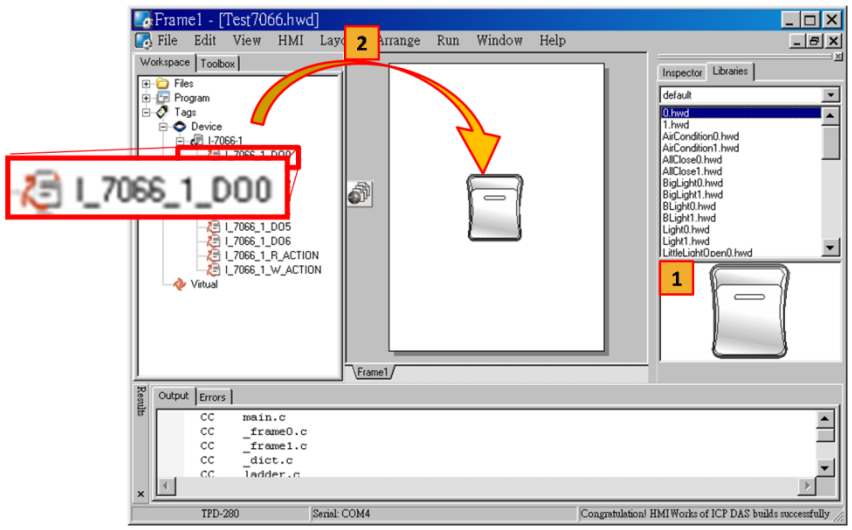
Here we just demonstrate how to quickly complete a whole new project with I/O modules of ICP DAS.

4. Designing the Ladder Diagram

Press **F3** on your keyboard or click the "**Register Devices**" option from the "**HMI**" menu to open the "**Devices**" window to register the I-7066 module.

Refer to the section "Connect to I/O Modules" for details.

Click the "**Libraries**" tab to select a picture to represent the tag in the "**Libraries**" panel. Drag and drop the tag that is corresponding to the DO0 of I-7066. On the frame design area, the picture you just select is now on the frame.



5. Setup Device

Refer to the section "Setup Devices" for details.

6. Compiling and Downloading to Run

After connecting to the TouchPAD device, press **F9** on your keyboard to run (or click the "**Run**" option from the "**Run**" menu).
As shown in the figure below, pressing the button switches the output of channel 0 of the I-7066 module.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

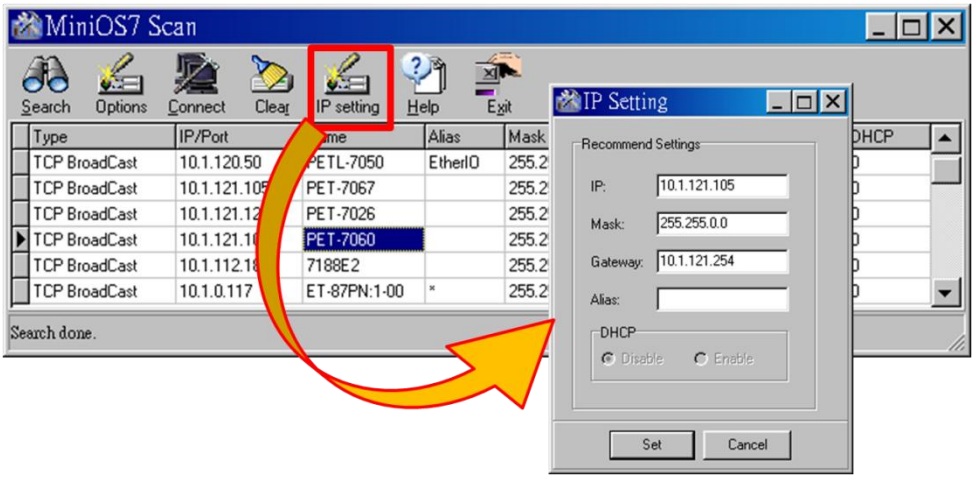
6.4. Integrating TPD-283 Series with I/O modules

In this example, we use the TPD-283 device to control a PET-7060 module, the 6-channel Power Relay Output, 6-channel Isolation Digital Input and PoE module of ICP DAS. First, put the PET-7060 module in the Ethernet network of the TPD-283 device and use a browser to configure the PET-7060 module.

1. Configuring PET-7066 by a Browser

Download the MiniOS7 Utility and its user manual from ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/minios7_utility/ Then use the MiniOS7 Utility to configure the IP settings of the PET-7060 module. (Be sure to make the PET-7060 module and your PC in the same subnet.) Press **F12** on your keyboard to scan through the network. Then click the PET-7060 module that is found and then click the "**IP setting**" button to configure the IP settings.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

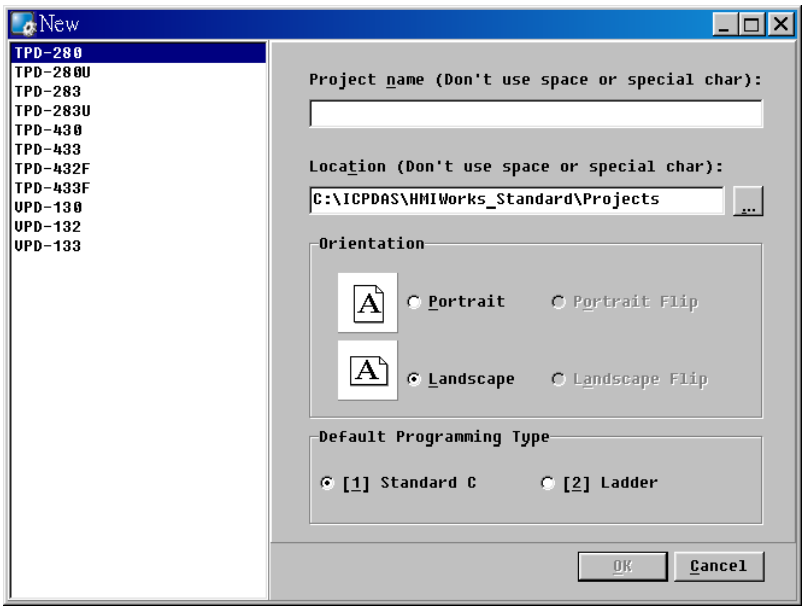


Finally, connect to the PET-7060 module and configure it by a browser.

2. Creating a New Project

Click the “**New…**” option from the “**File**” menu, and then select the Model, specify the Project name, the Location, the Orientation, and the Programming Type. Here we choose **programming type** as [2] Ladder.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



3. Designing the Graphic User Interface

We can skip this step.
Here we just demonstrate how to quickly complete a whole new project with I/O modules of ICP DAS.

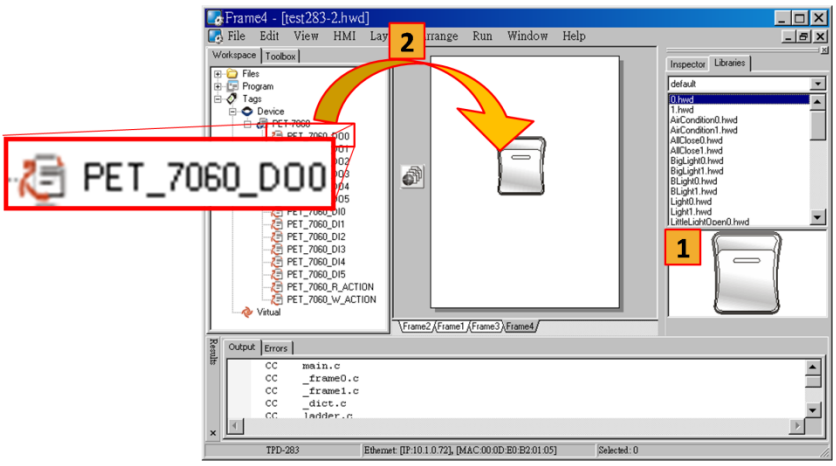
4. Designing the Ladder Diagram

Press **F3** in your keyboard or click the "**Register Devices**" option from the "**HMI**" menu to open the "**Devices**" window to register the PET-7060 module.

Refer to section "Connect to I/O Modules" for details.

Click the "**Libraries**" tab to select a picture to represent the tag in the "**Libraries**" panel. Drag and drop the tag that is corresponding to the DO0 of the PET-7060 module to the frame design area. On the frame design area, the picture you just select is now on the frame.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



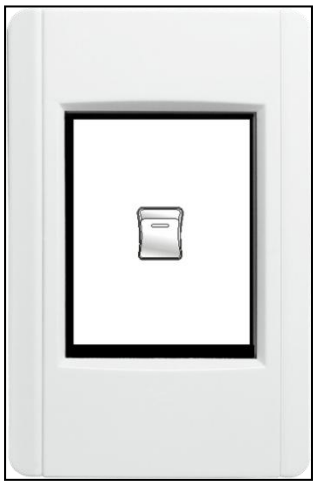
5. Setup Device

Refer to "Setup Devices" for details.

6. Compiling and Downloading to Run

After connecting to the TPD-283 device, press **F9** on your keyboard to run (or click the "**Run**" option from the "**Run**" menu).

As shown in the figure below, pressing the button switches the output of channel 0 of the PET-7060 module.



Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

7. Advanced Programming in C

We have an API reference for TouchPAD.
ftp://ftp.icpdas.com/pub/cd/touchpad/document/english/api_reference/

Though you can refer to the generated codes to learn how to use these API functions, all the API functions are defined in header files in the following path: “C:\ICPDAS\HMIWorks_Standard\include\grlib” and “C:\ICPDAS\HMIWorks_Standard\include”, where “C:\ICPDAS\HMIWorks_Standard” is the installation path.

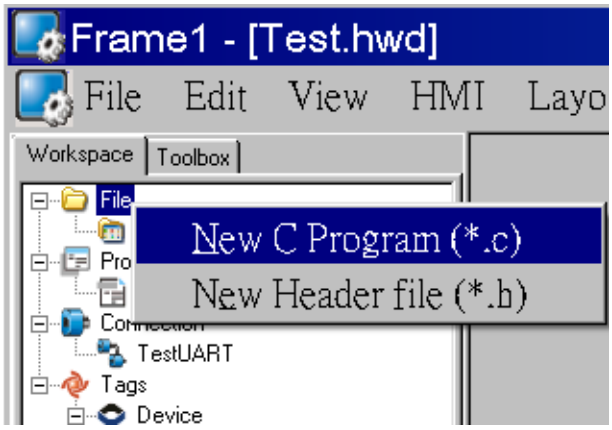
We give some examples in this chapter.

7.1. Adding a New File to Project

Before introducing the details, first we show how to add a new file (.c or .h) to the project.

1. Go to **Workspace**.
2. Right click on the “**File**” item and a pop-up menu is displayed.
3. On that pop-up menu, choose the type of the file you want to add.

As the following figure shows:



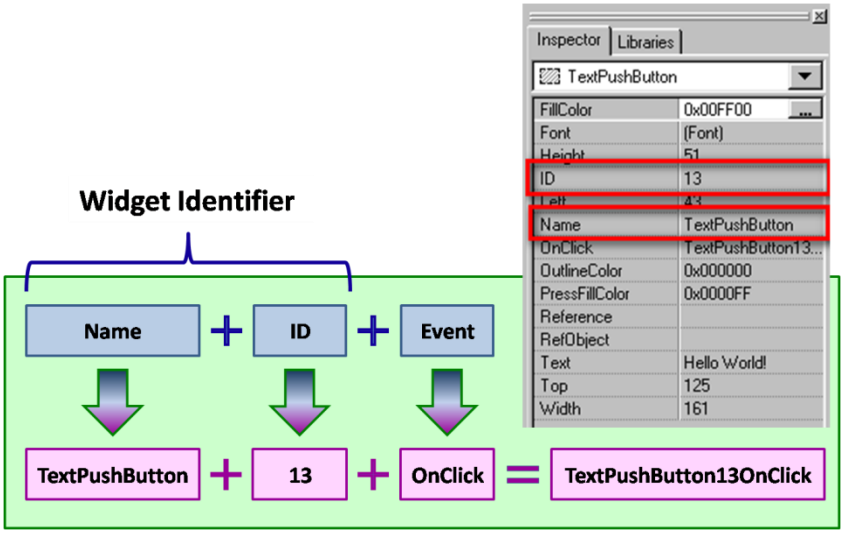
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

7.2. Updating Properties in Run Time

It is a bit more complicated to change the properties of widgets in the run time. In this section, we demonstrate some commonly-used cases, including:

1. The “FillColor” and “Text” properties of a TextPushButton component
2. The percentage of a Slider component
3. The “Selected” property of a CheckBox component
4. The “Font”, the “Text” and the “TextColor” properties of a Label component

Updating properties is implemented in the event handlers of the widgets.
Note: The naming convention of the event handler of the widget (here the widget is the TextPushButton component) is shown as below:



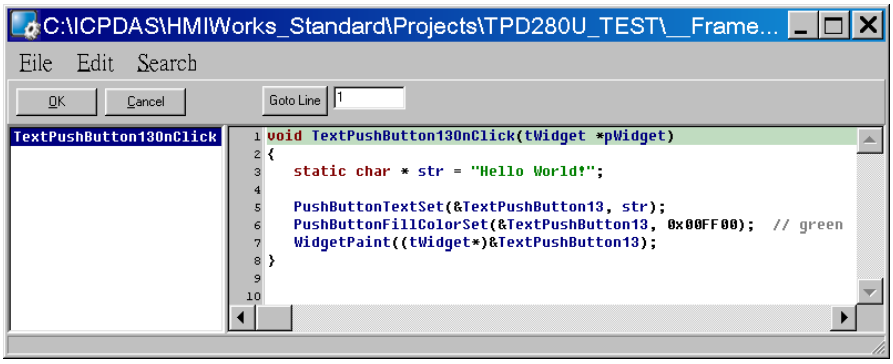
7.2.1. FillColor and Text of a TextPushButton

This section shows how to change the “FillColor” and the “Text” properties of a TextPushButton component. Simply follow the steps below.

1. Click the TextPushButton icon in the “**Toolbox**” panel and move your mouse to the frame design area. Click and drag a suitable sized TextPushButton.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

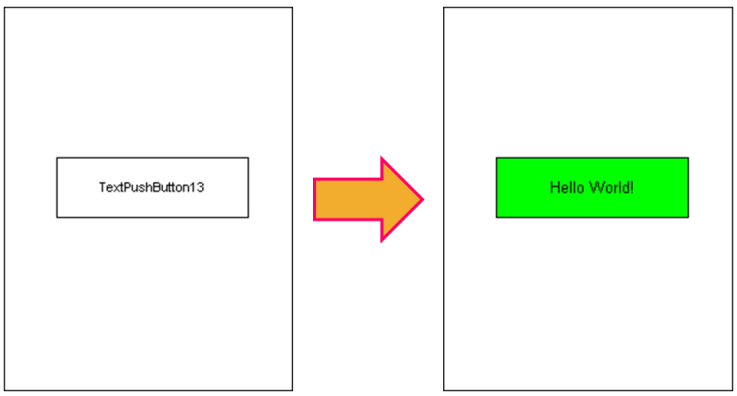
2. Double click the TextPushButton component to implement its OnClick event handler in the displayed programming window. Then press **OK** to save the file and leave.



3. In order to make it clearer, we copy the above codes below.

```
void TextPushButton13OnClick(tWidget *pWidget)
{
        static char * str = "Hello World!";

        PushButtonTextSet(&TextPushButton13, str);
        PushButtonFillColorSet(&TextPushButton13, 0x00FF00);    // green
        WidgetPaint((tWidget*)&TextPushButton13);
}
```

The effect of the OnClick event handler:



To set the “Text” property of a TextPushButton, we provide another function “**TextButtonTextSet**” for your convenience. Refer to the API reference for more details. The API reference can be downloaded from:

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

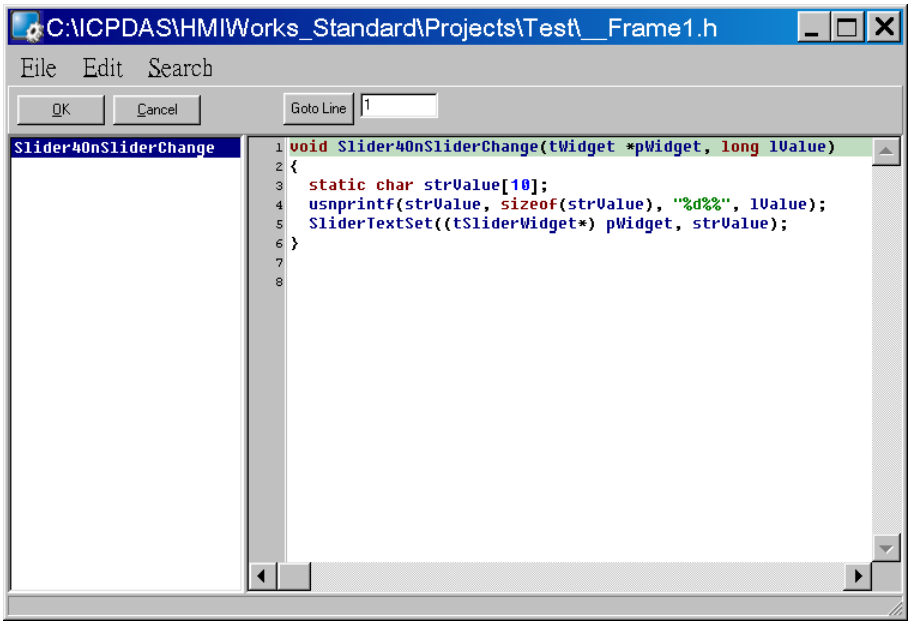
ftp://ftp.icpdas.com/pub/cd/touchpad/document/english/api_reference/

For more API functions of the TextPushButton component, refer to **pushbutton.h** in the following path:
“C:\ICPDAS\HMIWorks_Standard\include\grlib”, where
“C:\ICPDAS\HMIWorks_Standard” is the installation path.

7.2.2. Percentage of a Slider

Simply follow the steps below to display the percentage of a Slider when it changes its position.

1. Click the Slider icon in the “**Toolbox**” panel and move your mouse to the frame design area. Click and drag a suitable sized Slider.
2. Double click the Slider component to implement its OnSliderChange event handler in the displayed programming window. Then press **OK** to save the file and leave.



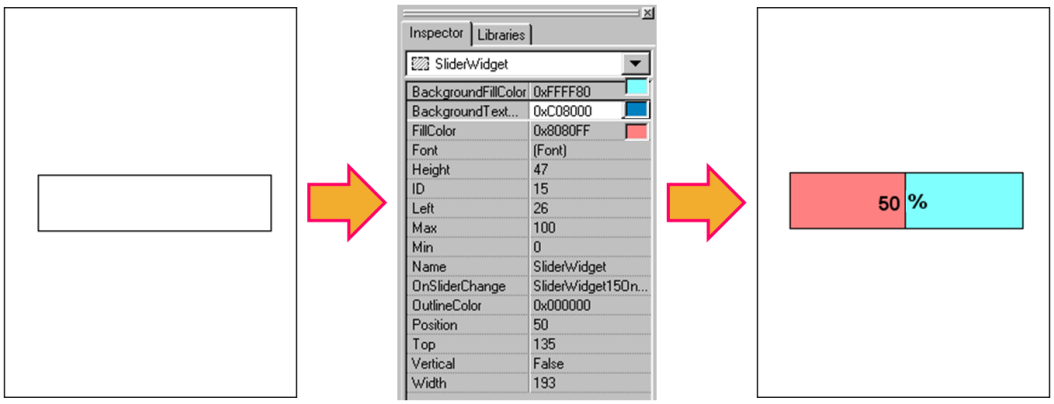
3. In order to make it clearer, we copy the above codes below.

```
void SliderWidget6OnSliderChange(tWidget *pWidget, long lValue)
{
   static char strValue[10];
```


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

```
    usnprintf(strValue, sizeof(strValue), "%d%%", lValue);
    SliderTextSet((tSliderWidget*) pWidget, strValue);
}
```

The effect of the OnSliderChange function (after selecting colors):



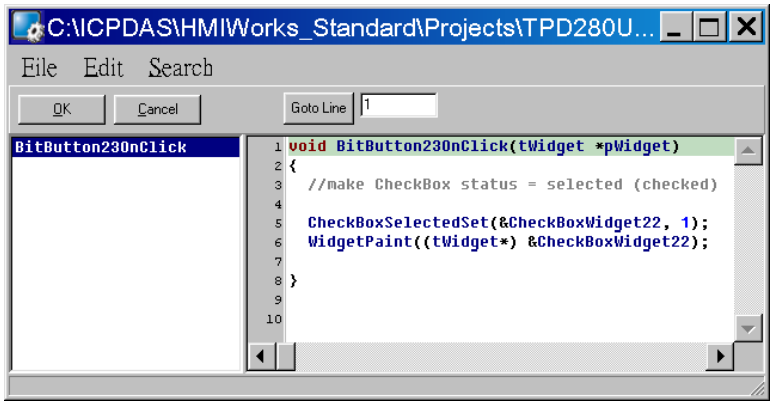
For more API functions of Slider, refer to **slider.h** in the following path: “C:\ICPDAS\HMIWorks_Standard\include\grlib”, where “C:\ICPDAS\HMIWorks_Standard” is the installation path.

7.2.3. Selected of a CheckBox

Take the steps below for example to change the “Selected” property of a CheckBox component in the run time.

1. Click the CheckBox icon in the “**Toolbox**” panel and move your mouse to the frame design area. Click and drag a suitable sized CheckBox.
2. Repeat the same procedure as that of the CheckBox component for a BitButton component.
3. Double click the BitButton component to implement its OnClick event handler in the displayed programming window. Then press **OK** to save the file and leave.

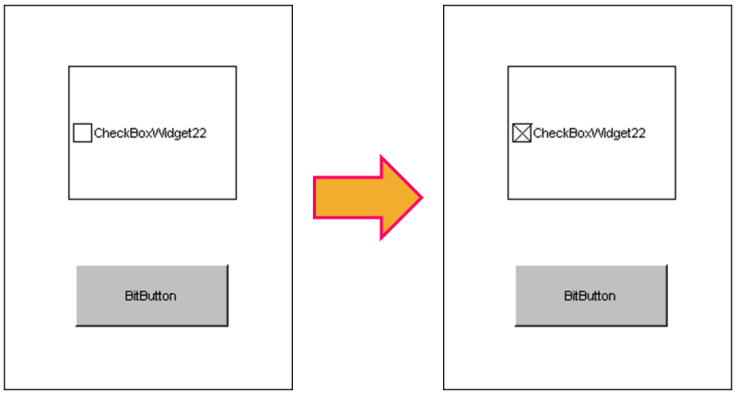
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com



4. In order to make it clearer, we copy the above codes below.

```
void BitButton23OnClick(tWidget *pWidget)
{
  //make CheckBox status = selected (checked)

  CheckBoxSelectedSet(&CheckBoxWidget22, 1);
  WidgetPaint((tWidget*) &CheckBoxWidget22);
}
```

The effect of the OnClick function:



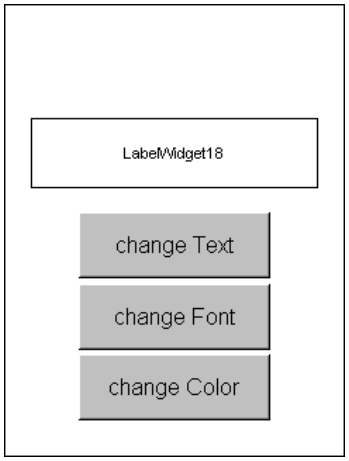
For more API functions of CheckBox, refer to **checkbox.h** in the following path:
“C:\ICPDAS\HMIWorks_Standard\include\grlib”, where
“C:\ICPDAS\HMIWorks_Standard” is the installation path.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

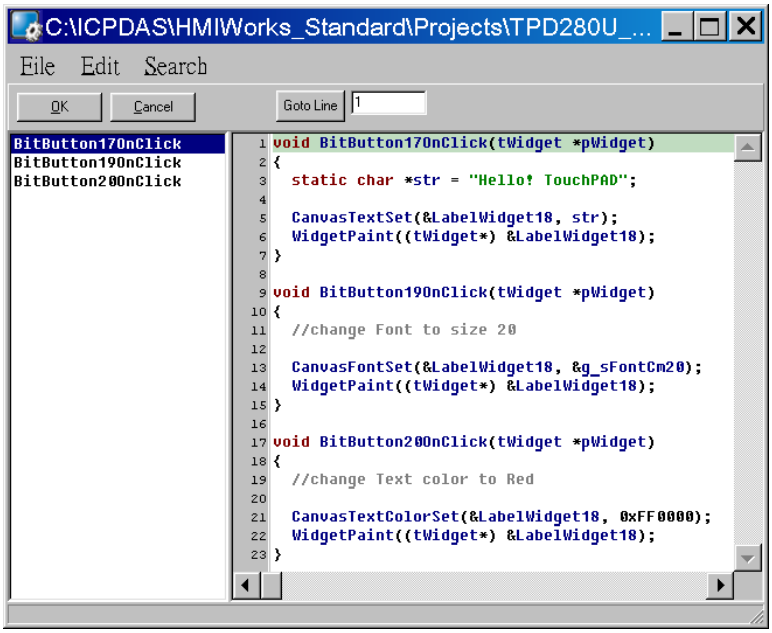
7.2.4. Font, Text and TextColor of a Label

Take the steps below for example to update properties of a Label component in the run time.

1. Click the Label icon in the "**Toolbox**" panel and move your mouse to the frame design area. Click and drag a suitable sized Label.
2. Repeat the same procedure as that of the Label component above for three BitButton components.



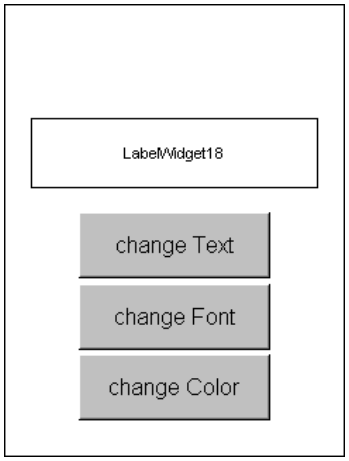
3. Double click the BitButton component to implement its OnClick event handler in the displayed programming window. Then press **OK** to save the file and leave.

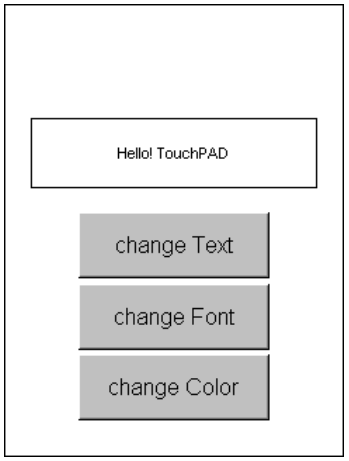


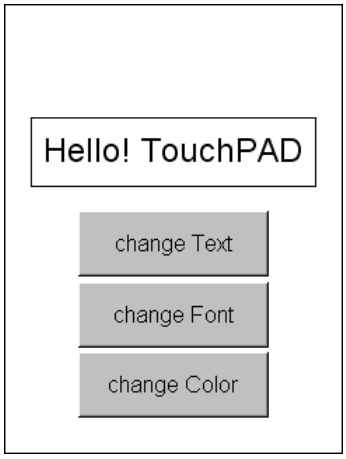
Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

4. In order to make it clearer, we copy the codes below with their corresponding results.

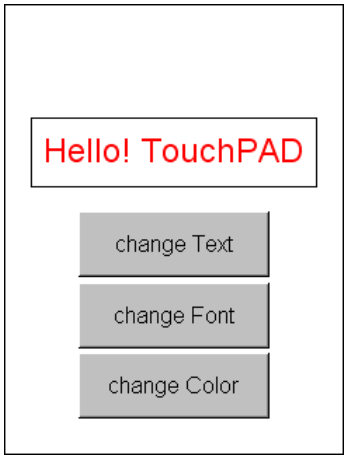
Results	Codes of the event handler

	//step 0
	//the beginning snapshot
	N/A

	//step 1
	//Click on BitButton “change Text”
	
	void BitButton17OnClick(tWidget *pWidget)
	{
	  static char *str = "Hello! TouchPAD";
	
	  CanvasTextSet(&LabelWidget18, str);
	  WidgetPaint((tWidget*) &LabelWidget18);
	// or use **LabelTextSet** to replace
	// **CanvasTextSet** and **WidgetPaint**, that is:
	// **LabelTextSet**(&LabelWidget18, str);
	}

	//step 2
	//Click on BitButton “change Font”
	
	void BitButton19OnClick(tWidget *pWidget)
	{
	  //change Font to size 20
	
	  CanvasFontSet(&LabelWidget18, &g_sFontCm20);
	  WidgetPaint((tWidget*) &LabelWidget18);
	}

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com




```
//step 3
//Click on BitButton “change Color”

void BitButton20OnClick(tWidget *pWidget)
{
   //change Text color to Red

   CanvasTextColorSet(&LabelWidget18,
0xFF0000);
   WidgetPaint((tWidget*) &LabelWidget18);
}
```

To set the “Text” property of a Label component, we provide another function “**LabelTextSet**” for your convenience. Refer to the API reference for more details. The API reference can be downloaded from: ftp://ftp.icpdas.com/pub/cd/touchpad/document/english/api_reference/

For more API functions of Label, refer to **canvas.h** in the following path: “C:\ICPDAS\HMIWorks_Standard\include\grlib”, where “C:\ICPDAS\HMIWorks_Standard” is the installation path.

In the same path, there is a header file, grlib.h.
grlib.h contains prototypes for the pre-defined fonts, such as g_sFontCm20.

7.3. Accessing Tags in Ladder

In HMIWorks, users can design a project with many frames of two different types, “Standard C” and “Ladder”. The variables (tags) used in the Ladder is transformed into a structure of the C language after building the project and thus the tags can be accessed in the frame of programming type “Standard C”.

Two macros are provided for this purpose:

1. **VAR_GET**: get the value from the tag in the Ladder
2. **VAR_SET**: set a value to the tag in the Ladder

Supposed that we have a tag named “count” incremented in the Ladder, then

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

we can get the value of the “count” tag and set the “count” tag to zero as shown in the example below.

```
static char str[32];

//get count (virtual tag) from Ladder
void TextPushButton8OnClick(tWidget *pWidget)
{
   VAR_GET(count);

   usprintf(str, "%d", count);
   LabelTextSet(&LabelWidget9, str);
}

//Set count to zero
void TextPushButton11OnClick(tWidget *pWidget)
{
   VAR_SET(count, 0);

   usprintf(str, "%d", count);
   LabelTextSet(&LabelWidget9, str);
}
```


Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Appendix

Appendix is listed below:

1. FAQ, Frequently Asked Questions

A.FAQ

A.1. What to do if screen flashes?

Go to the section “Properties of a Frame” for details.

A.2. How to have higher-resolution Picture?

Go to the section “Loading a Picture” for details.

A.3. How does a TouchPAD control I/O?

Go to the section “Using an ObjectList” for details.

A.4. How to change Font of Text?

Go to the section “Drawing a Text” for details.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

A.5. How to represent decimals for Ladder Designer?

Go to the section “Using a Label” for details.

A.6. How to remove the startup beep of a TPD-283 device?

The TPD-283 devices sound a beep when startup.
Refer to the section “The Options of TouchPAD” for more details.

A.7. How to customize the generated code?

Every time when building a project, HMIWorks generates source codes to build. Below is the procedure to customize the generated source codes.

1. After finishing designing the project, press **F5** (build) on your keyboard instead of **F9** (run) to generate codes.
2. In the directory of the project, open the source file (.c files).
3. Edit the source files (.c files).
4. Press **F10** on your keyboard, and a “cmd.exe” window is displayed. Enter “**make**” in the “cmd.exe” window to re-make the project.
5. For the TPD-283U-H/TPD-283U-Mx, there are additional steps that need to be executed after entering **make**.
   Enter “make genbix”
6. Click the “**Download Only** (**Ctrl** + **F9**)” option from the “**Run**” menu to download the .bin (or .bix) file.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

A.8. How to store data in the flash?

For users' convenience, there are two sets of API functions for data storage in the flash on the TouchPAD devices. One is for the MCU (micro-controller unit) internal flash and the other is the external serial flash.
To user these features, install the HMIWorks software with version 2.03 or above.
ftp://ftp.icpdas.com/pub/cd/touchpad/setup/

No.	1	2
Target Flash	MCU internal flash	External serial flash
Possible Target Device	All devices in the TouchPAD series	All devices in the TouchPAD series, except TPD-280 and TPD-283 (for those having external flash)
API Functions Provided*	hmi_UserParamsGet, hmi_UserParamsSet	hmi_UserFlashReadEx, hmi_UserFlashWriteEx, hmi_UserFlashConfig, hmi_UserFlashErase
Size of Storage	256 byte	4 KB ~ 7 MB
Suggested Users	Any TouchPAD users	**For advanced users only.** Any undetermined use will damage the application image.

* Refer to the API reference for more details.
ftp://ftp.icpdas.com/pub/cd/touchpad/document/english/api_reference/

A.9. How to use soft reset?

There are two methods to reset a TouchPAD by software.

Method 1:

Use the API function of hmi_SoftwareReset.

Method 2:

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com

Use the Watchdog.

Step 1: Enable watchdog.

Go to HMI->Project Configuration, and then enable the watchdog option.

Step 2: Use infinite loop to start up watchdog.

For example: while(1){}

If you need to use this function in ladder, Refer to the section “User-Dfined Function Block” for more details.

Copyright © 2015 ICP DAS Co., Ltd. All Rights Reserved. ✉ E-mail: service@icpdas.com